Deep learning-based methods for defect detection from ultrasound images

Medak, Duje

University of Zagreb

FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

Duje Medak

DEEP LEARNING-BASED METHODS FOR DEFECT DETECTION FROM ULTRASOUND IMAGES

DOCTORAL THESIS

Zagreb, 2022

University of Zagreb

FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

Duje Medak

DEEP LEARNING-BASED METHODS FOR DEFECT DETECTION FROM ULTRASOUND IMAGES

DOCTORAL THESIS

Supervisor: Professor Sven Lončarić, PhD

Zagreb, 2022

Sveučilište u Zagrebu

FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Duje Medak

METODE ZASNOVANE NA DUBOKOM UČENJU ZA DETEKCIJU DEFEKATA IZ ULTRAZVUČNIH SLIKA

DOKTORSKI RAD

Mentor: Prof. dr. sc. Sven Lončarić

Zagreb, 2022.

The doctoral thesis was completed at the University of Zagreb Faculy of Electrical Engineering and Computing, Department of Electronic Systems and Information Processing.

Supervisor: Professor Sven Lončarić, PhD

The thesis has 107 pages.

Thesis number: _____

About the Supervisor

Dr. Sven Lončarić is a distinguished professor at the Faculty of Electrical Engineering and Computing, University of Zagreb. He received a Diploma of Engineering and Master of Science degrees in electrical engineering from the University of Zagreb, Faculty of Electrical Engineering and Computing in 1985 and 1989, respectively. As a Fulbright scholar, he received a Ph.D. degree in the field of image processing and analysis from the University of Cincinnati, OH in 1994. He works at the Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia since 1996. He first worked there as an assistant professor (1996), from 2001 he worked as an associate professor, from 2006 as a full professor, and from 2011 he works as a distinguished professor. He was also an assistant professor from 2001 to 2003 at the New Jersey Institute of Technology, Newark, USA. His areas of research interest are image processing and computer vision. Together with his students and collaborators, he published more than 200 publications in scientific peer-reviewed journals and at international conferences. Prof. Lončarić is the founder and head of the Center of Excellence for Computer Vision at the University of Zagreb. He is the head of the research laboratory for image processing at the Faculty of Electrical Engineering and Computing and the President of the Committee for the Development of Biomedical Engineering at the University of Zagreb. He is a Co-Director of the national Center of Excellence for Data Science and Cooperative Systems. He is a senior member of IEEE and a member of the Croatian Academy of Technical Sciences. Prof. Lončarić received many awards for his scientific and professional work including the National Science Award for outstanding scientific achievements from the Croatian Parliament, Golden Plaque "Josip Lončar", "Fran Bošnjaković" award, and "Rikard Podhorsky" award. He was featured on a comprehensive list of the top 1% world's scientists in the category "Artificial intelligence and image processing" released in 2021 by Stanford University.

O mentoru

Dr. sc. Sven Lončarić je redoviti profesor u trajnom zvanju na Fakultetu elektrotehnike i računarstva na Sveučilištu u Zagrebu. Diplomirao je i magistrirao na Elektrotehničkom fakultetu (ETF) u Zagrebu 1985. odnosno 1989. godine. Bio je dobitnik Fulbright-ove stipendije i doktorirao je na 1994. godine na Sveučilištu u Cincinnatiju, SAD u području obrade i analize slike. Od 1996. godine docent je na Fakultetu elektrotehnike i računarstva (FER) u Zagrebu. Od 2001 do 2003. godine bio je profesor na sveučilištu New Jersey Institute of Technology, Newark, SAD. Od 2001. godine je izvanredni profesor na FER-u, od 2006. godine redoviti profesor, a od 2011. godine redoviti profesor u trajnom zvanju u području elektrotehnike i u području računarstva. Područja njegovog istraživačkog interesa su obrada slike i računalni vid. Sa svojim

doktorandima i suradnicima objavio je više od 200 znanstvenih i stručnih radova u međunarodnim časopisima i na skupovima. Osnivač je i voditelj Centra izvrsnosti za računalni vid na Sveučilištu u Zagrebu. Voditelj je istraživačkog laboratorija za obradu slike na FER-u. Predsjednik je Odbora za razvoj biomedicinskog inženjerstva na Sveučilištu u Zagrebu. Suvoditelj je nacionalnog Znanstvenog centra izvrsnosti za znanost o podatcima i kooperativne sustave. Stariji je član IEEE i redoviti član Akademije tehničkih znanosti Hrvatske. Prof. Lončarić je dobitnik mnogih nagrada za znanstveni i stručni rad uključujući Državnu nagradu za znanost koju mu je Hrvatski sabor dodijelio 2019. godine, Zlatnu plaketu "Josip Lončar" FER-a, Nagradu "Fran Bošnjaković" Sveučilišta u Zagrebu i Nagradu "Rikard Podhorsky" Akademije tehničkih znanosti Hrvatske. Prema listi koju je Sveučilište Stanford objavilo 2021. godine, prof. Lončarić nalazi se među top 1% svjetskih znanstvenika u kategoriji "Umjetna inteligencija i obradba slike".

Preface

This thesis is based upon the results of the research conducted from 2018 to 2022 as a part of the project "Smart UTX" (KK.01.2.1.01.0151) co-financed by the INETEC d.o.o. and the European Union from the European Regional Development Fund. The project was supervised by Professors Sven Lončarić, PhD, Marko Subašić, PhD and Tomislav Petković, PhD.

I would like to express sincere gratitude to my mentor Professor Sven Lončarić for his guidance and continuous support during my doctoral study, and for providing me with opportunities to work on many interesting and challenging problems that helped me to acquire new skills and knowledge. I would also like to thank Professor Marko Subašić, with whom I had many fruitful discussions and gain valuable advice regarding my research.

At times, the goal of finishing a PhD seemed unreachable and distant, and the path looked difficult. Despite this, when I think about the last years, I realize it was also a lot of fun. For this, I am grateful to my ZESOI colleagues and friends, especially Luka, who showed me how to go through a PhD (and life) with a smile on my face. To all of my friends with whom I shared my struggles (mostly in Desirée), thank you. To my girlfriend Bea, that celebrated my accomplishments often more than myself, and was my biggest support throughout the whole doctoral study, thank you. Finally, to my family, my brother Marin, my father Tomislav, and especially my mother Gorana I am eternally grateful for the love and support you gave me throughout my whole education.

Abstract

Components of many systems, structures, and buildings, require constant monitoring and inspections because of possible defect occurrences due to constant usage and material stress. To inspect the material and prevent component failure, a wide range of non-destructive evaluation (NDE) techniques can be applied. Ultrasonic testing (UT) is one of the NDE techniques that is commonly used today due to its many advantages. UT is quite simple to employ since only one-sided access to the material is needed, and the internal structure of a material can be inspected with the ability to precisely localize defects within the inspected component. Acquisition of UT data is nowadays mostly performed in an automated fashion, using the robotic manipulator. The manipulator moves the ultrasonic transducer along the surface of the material. At each position, the ultrasonic probe (transducer) transmits and receives ultrasonic waves. In case of a defect presence, a fraction of the transmitted waves will bounce off the defect back to the probe, and by analyzing the received signal it is possible to precisely determine the defect's position and size. The analysis of the acquired data is currently done manually, making the process heavily reliant on the personnel's previous experience and knowledge. Manual analysis of the data can lead to error, especially when a large amount of data needs to be inspected and the repetitive work leads to fatigue of the inspectors. To overcome these problems, many researchers have proposed methods for the automated analysis of UT data. The main problem with the automated analysis is the irregularity of the acquired data, which makes it impossible to write an algorithmic description of the analysis process as done by the human inspector. In recent years, deep learning-based approaches emerged as one of the promising directions in the development of automated UT data analysis solutions. Deep learning approaches can implicitly learn the important features from the large datasets of labeled data. While in some cases it is possible to apply existing deep learning architectures for the analysis of UT data, some domain-specific challenges occur and limit the performance of such methods. For example, extreme aspect ratios of the defects in ultrasonic images limit the precision that can be achieved by the existing one-stage object detectors. Furthermore, when detecting a defect on an ultrasonic image, it would be useful to use additional information available from the surrounding area but a method that simultaneously processes several ultrasonic images was not yet proposed in the literature. In this thesis, several solutions and novel architectures are proposed in order to solve the aforementioned challenges. All of the proposed methods were tested on an in-house dataset with over 4000 ultrasonic B-scans. Experimental results confirm that the precision can be significantly improved by developing a novel deep learning architecture specifically designed for defect detection from ultrasound images.

Keywords: ultrasound image analysis, non-destructive evaluation, automated defect detection, object detection, data augmentation, image generation, deep learning

Prošireni sažetak

Metode zasnovane na dubokom učenju za detekciju defekata iz ultrazvučnih slika

Nerazorno ispitivanje je skup tehnika koje se upotrebljavaju za inspekciju materijala ili dijela nekog sustava bez nanošenje štete ispitivanoj komponenti. Brojne takve tehnike su razvijene tijekom godina i često se koriste prilikom inspekcije elektrana, zrakoplova, cjevovoda i sličnih konstrukcija gdje je nužno na vrijeme detektirati defekte. Neke od metoda nerazornog ispitivanja su metoda vrtložnih struja, vizualne metode, radijacijske metode, toplinske metode te ultrazvučno testiranje. Nekada se koriste i kombinacije različitih metoda kako bi se povećala pouzdanost inspekcije. Ultrazvučno testiranje (UT) ističe se među nabrojanim metodama zbog brojnih prednosti. Za početak, dovoljan je pristup samo jednoj strani materijala, a metoda svejedno daje uvid u internalnu strukturu i stanje materijala. Ultrazvučnim testiranjem se uglavnom dobiju podaci s visokim omjerom signala i šuma što omogućuje preciznu lokalizaciju defekta i određivanje njegovih dimenzija. Ultrazvučno testiranje bazira se na generiranju i detekciji ultrazvučnih valova unutar testnog objekta. Ako je defekt prisutan u materijalu, njegova gustoća se razlikuje od okolnog područja pa će to uzrokovati odbijanje dijela ultrazvučnih valova. Sonda će registrirati reflektirane ultrazvučne valove te se iz informacija o svojstvima materijala može izračunati točna dubina na kojoj se defekt nalazi. Dio odaslanih ultrazvučnih valova se također odbija od nepravilnosti u materijalu zbog čega se pojavljuje šum. Kako bi se povećala pouzdanost pronalaska defekta, prikupljanje ultrazvučnih podataka se danas uglavnom obavlja korištenjem sondi s faznim poljima (engl. phased array). Sonde s faznim poljima istovremeno odašilju ultrazvučne valove pod raznim kutovima (npr. od 45° do 79° s rezolucijom od 2°). Korištenjem ovog tipa sondi, smanjuje se vjerojatnost da se valovi neće odbiti od plosnatog defekta postavljenog paralelno u odnosu na putanju ultrazvučnih valova. Problem je što se količina podataka povećava korištenjem ovog tipa sonde pa je za analizu ovakvih podataka potrebno puno vremena. Prikupljeni ultrazvučni podaci se mogu prikazati u raznim formatima. Najjednostavniji prikaz se zove A-sken i on pokazuje količinu primljene energije kao funkciju vremena (ili dubine). Jedan A-sken se dobije kada sonda generira i primi jedan ultrazvučni val. Kontinuiranim pomicanjem sonde po površini materijala dobije se niz A-skenova. Niz A-skenova uglavnom se prikazuje u obliku slike koja se zove B-sken. B-sken se dobije pretvorbom amplituda A-skena u vrijednosti piksela. Kako bi se ispitao cjelokupni volumen materijala, sonda se pomakne u stranu svaki put se prikupi jedan B-sken. Na ovaj način se tijekom inspekcije prikupi niz B-skenova pri čemu svaki B-sken odgovara određenom presjeku materijala. Uz spomenute, postoje i brojni drugi načini prikaza prikupljenih podataka (C-sken, S-sken, itd.), ali ovi spomenuti su najčešće korišteni. Inspektori ručno pregledavaju prikupljene podatke kako bi utvrdili eventualnu prisutnost defekata u materijalu. Inspektori pritom istovremeno gledaju u razne prikaze podataka kako bi potvrdili svoju odluku. Često je za ispravnu odluku potrebno

pogledati i okolna područja oko sumnjive lokacije ili tu istu lokaciju pogledati pod drugim kutom. Količina podataka koja se prikupi tijekom stvarne inspekcije je ogromna zbog čega je analiza UT podataka jako zamorna i teška. Nadalje, većina slika uopće ne sadrži defekte tako da inspektori većinu vremena provode gledajući u monotone podatke. Ovaj postupak je jako repetitivan i naporan za ljude pa zbog umora može doći do pogreške u analizi podataka odnosno ne primjećivanja defekta. Automatizirani sustav bi mogao ovaj zadatak izvoditi puno brže, a dobiveni rezultati bi bili konzistentni. Bilo bi dovoljno da takav sustav pronađe sumnjive dijelove podataka, a ljudski inspektor bi zatim mogao pregledati taj manji izdvojeni dio i provesti daljnju analizu po potrebi.

Puno truda je uloženo u razvoj metoda koje bi mogle asistirati inspektorima prilikom analize UT podatka. Rani pokušaju se uglavnom oslanjaju na ekstrakciju značajki valićnom transformacijom i klasifikacijom ekstrahiranih značajki korištenjem strojnog učenja. Ovakvim pristupom se za svaki A-sken utvrdi sadrži li on signal defekta ili ne. Kao klasifikator se najčešće koriste umjetne neuronske mreže ili stroj potpornih vektora. Informacije iz A-skena se mogu izvući i korištenjem drugih transformacija ili kombinacijom raznih transformacija. Neki autori se za cjelokupni proces analize oslanjaju na nadzirano učenje korištenjem umjetnih neuronskih mreža. U tom slučaju potrebno je imati dovoljno veliki skup podataka iz kojega model onda može implicitno naučiti bitne značajke i na temelju njih razlikovati defektne od normalnih A-skenova. Za ovakav pristup posebno je popularna specijalna vrsta neuronske mreže koja se zove konvolucijska neuronska mreža (engl. convolutional neural network). Konvolucijske neuronske mreže mogu direktno iz podataka naučiti koje informacije su bitne pa se ne treba provoditi ručno dizajnirana ekstrakcija značajki. Konvolucijske neuronske mreže su pogotovo efikasne prilikom analize jednodimenzionalnih ili dvodimenzionalnih struktura podataka kao što su sekvence ili slike. Bez obzira na način analize pojedinog A-skena, često je teško provesti klasifikaciju bez da se u obzir uzmu i okolni A-skenovi. Glavi razlog je sličnost signala uzrokovanih geometrijom komponente ili šumom i signala nastalog refleksijom od defekta.

Zbog toga se osim metoda za analizu A-skenova, razvijaju i metode za analizu B-skenova. Dugo vremena metode za analizu slika nisu bile dovoljno razvijene kako bi se uspješno detektirali defekti na B-skenovima. Situacija se nedavno poboljšala razvojem raznih arhitektura dubokih neuronskih mreža, te procedura korištenih za njihovo treniranje. Postoji mnogo javno dostupnih skupova slika na kojima se testiraju generalne sposobnosti predloženih arhitektura za razne zadatke kao što su klasifikacija slike, detekcija i praćenje objekata na slikama, semantička segmentacija, itd. Tijekom godina se poboljšala efikasnost predloženih arhitektura te je mnogim predloženim tehnikama kao što su augmentacija podataka i prijenosno učenje, omogućena primjena postojećih modela u novim domenama. Posljedično se povećao i broj radova koji duboke neuronske mreže upotrebljavaju za analizu B-scanova dobivenih ultrazvučnim testiranjem. Ako je dostupan dovoljno veliki skup slika, za detekciju defekata mogu su primijeniti postojeći de-

tektori koji se često dijele u dvije obitelji: jednofazni (engl. one-stage) i dvofazni (engl. two-stage). Dvofazni detektori su u vrijeme pojavljivanje postizali bolje rezultate od jednofaznih, ali su bili sporiji. Kao što i samo ime nalaže kod njih se detekcije provodi u dvije faze. U prvoj fazi se na slici identificiraju područja koja potencijalno sadrže objekte od interesa. U Drugoj fazi se odbacuju područja za koja se odredi da ne sadrže objekte, a za preostale objekte se izračuna njihova točna lokacija određena graničnim okvirom. Gledano na nekoj apstraktnoj razini, može se reći da su dvofazni detektori nastali iz tradicionalnog pristupa gdje se prvo nekakvim segmentacijskim algoritmom poput selektivnog pretraživanja odrede regije od interesa, a zatim se provodi klasifikacija. Ovakav pristup je pogotovo prikladan kada se analiziraju slike u kojima su prikazane kompleksne pozadine ili objekti. Takve slike su uobičajene u javno dostupnim skupovima slika koji se koriste za evaluaciju novih metoda, ali kod ultrazvučnih slika to nije slučaj. Jednofazni detektori detekciju provode u jednoj fazi, koristeći gustu mrežu preddefiniranih oblika za koje model pokušava utvrditi pripadaju li nekom od objekata koje je potrebno detektirati. Kada se za neki od predodređenih oblika, koji se još u literaturi nazivaju i sidreni okviri (engl. anchor boxes, priors, or default boxes), zaključi da sadrži objekt, njegov oblik se dodatnom transformacijom modificira tako da bolje enkapsulira objekt kojeg je potrebno detektirati. Jednofazni detektori su u početku bili brži od dvofaznih, ali manje precizni. Razlika u preciznosti s vremenom je nestala, a danas je većina novo predloženih suvremenih detektora objekata (EfficientDet, YOLOv5) jednofaznog tipa.

Postojeće detektore objekata moguće je primijeniti za detekciju defekata na B-skenovima, ali će zbog raznih problema koji se pojavljuju preciznost takvih pristupa biti ograničena. Neki od problema koji se pojavljuju su mali skup slika za treniranje i evaluaciju, šumovite slike na kojima je teško razlikovati signal defekta od signala uzrokovanog geometrijom ili šumom te ekstremni omjeri defekata. Prvi problem utječe na veličinu modela kojeg je moguće istrenirati. U običajnim situacijama, ako imamo dovoljno veliki skup slika moguće je povećavati model dodavanjem novih slojeva i povećavanjem broja filtera sve dok model ne dosegne dovoljan kapacitet za uspješno obavljanje zadatka. Ako je skup podataka manji, potrebno je smanjiti i model, kako bi se svi parametri (težine) modela mogle naučiti. Korištenje kompleksnih modela, čak i kada je dostupna baza slika mala, donekle je moguće kada se koristi tehnika prijenosnog učenja (engl. transfer learning) i augmentacija podataka uz pomoć koje umjetno povećamo broj dostupnih slika. Ovi pristupi međutim i dalje imaju ograničenja i nerealno je očekivati da se desetci milijuna parametara uspješno mogu izračunati iz ograničenog broja primjera za učenje. Jedno od rješenja ovog problema je izrada novog modela s manje parametara od modela koji se učestalo koriste za zadatke na javno dostupnim skupovima slika i koji su namijenjeni za treniranje na milijunima slika. Dobrim dizajniranjem nove arhitekture potencijalno se može dobiti veća preciznost i mogućnost treniranje čak i kada je baza podataka mala. Model i dalje mora imati dovoljan kapacitet kako bi mogao raditi sa šumovitim podacima što je prije u radu istaknut

kao drugi problem. Treći problem je da se tijekom treniranje detektora objekata baziranih na dubokom učenju koriste samo sidreni okviri koji dobro enkapsuliraju objekte koje je potrebno detektirati. Zbog toga je namještanje hiperparametara sidrenih okvira jako bitan i težak zadatak. Predloženi oblici sidrenih okvira računaju se iz definiranih omjera i faktora skaliranja. Iako za neke modele postoje algoritmi za izračun tih vrijednosti, za neke suvremene detektore to je pitanje otvoreno i još nije jednoznačno određen najbolji način za postavljanje tih vrijednosti. Ako početne vrijednosti nisu dobro postavljene, treniranje je otežano a krajnja preciznost detektora limitirana.

Na početku ovog doktorskog rada, napravljena je usporedba suvremenih detektora objekata i uspoređene su njihove performanse. Predložena je nova procedura za izračun hiperparametara sidrenih okvira kod EfficientDet arhitekture. Eksperimentima je potvrđeno da se korištenjem sidrenih okvira dobivenih predloženom procedurom ostvaruje značajno veća preciznost. U idućoj iteraciji je predložena potpuno nova arhitektura. Implementiran je novi ekstraktor značajki uz pomoć kojega se postižu još bolji rezultati i to uz značajno ubrzanje u usporedbi s prethodnom arhitekturom. Novi detektor ima i modificiranu glavu za detekciju koja je dizajnirana specifično za detekciju objekata s ekstremnim omjerima stranica. Nova arhitektura na ispitnoj bazi podataka ultrazvučnih B-skenova ostvaruje veću preciznost od svih drugih testiranih arhitektura. Iako je utjecaj prethodno navedenih problema minimiziran implementacijom ove arhitekture , preciznost detektora koji pojedinačno analizira B-skenove i dalje je djelomično ograničena. Nekada je jednostavno nemoguće razaznati je li neki signal nastao odbijanjem od defekta ili pak zbog odbijanja od geometrije ispitivane komponente ili šuma. Kada ljudski inspektori provode analizu, njihova odluka ovisi i o okolnim područjima (susjedni B-skenovi) ili o prikazima istog područja dobivenima snimanjem pod drugim kutom (najčešće korištenjem sonde s faznim poljima). Kako bi se dodatno poboljšala preciznost detektora defekata, osmišljeno je i implementirano nekoliko novih arhitektura za istovremenu analizu više ultrazvučnih B-skenova.

Jedna od predloženih arhitektura se koristi za ubrzanje automatske analize u realnim situacijama gdje je potrebno analizirati podatke dobivene skeniranjem metalnog bloka sondom s faznim poljima. Predložena arhitektura, uz minimalne gubitke preciznosti, istovremeno analizira slike pod svim kutovima. To je izvedeno tako što se prvo provede dinamičko spajanje slika, a zatim detektira defekte u rezultantnoj slici. Težina pojedine ulazne slike, određuje se korištenjem submodela koji uz pomoć nekoliko 3D konvolucijskih slojeva i mehanizmom pažnje određuje važnost pojedine ulazne slike.

Druga predložena arhitektura za analizu sekvenci ultrazvučnih slika, koristi se za poboljšanje preciznosti detektora. Poboljšanje je ostvareno proširenjem ulaza u model u 3D volumen. Umjesto analize jednog B-skena, ovaj model analizira sekvencu B-skenova koji prikazuju susjedne presjeke materijala. Za početak je pokazano da naivan pristup gdje se ulaz u postojeće detek-

tore samo proširi iz trokanalne slike u devet-kanalnu sliku ne dovodi do poboljšanja. Zatim su predložena dva nova pristupa koja su bazirana na izračunu značajki iz pojedine ulazne slike, njihovom spajanju u visokodimenzionalnom prostoru značajki te provođenju detekcije iz dobivenih značajki. Za spajanje značajki su isprobana dva pristupa, jedan baziran na običnom dvodimenzionalnom konvolucijskom sloju te jedan baziran na konvolucijskom LSTM (Long-short term memory) sloju. Eksperimentalno je pokazano da predloženi pristupi dodatno povećavaju preciznost detektora defekata.

Ključne riječi: analiza ultrazvučnih slika, nerazorno ispitivanje, automatska detekcija defekata, detekcija objekata, augmentacija slike, generiranje slika, duboko učenje

Contents

Chapter 1

Introduction

Non-destructive evaluation is a set of techniques used to examine objects of any type, size, shape, or material to determine the presence or absence of discontinuities such as defects, or to evaluate other material characteristics [1, 2]. Applying an NDE method does not cause damage to the inspected component, and it does not affect its usability. This property makes NDE methods perfect for continuous inspection of critical components in many systems, especially if the inspected component is expensive to manufacture. Some examples of NDE techniques include:

- •Visual testing (VT)
- •Penetrant testing (PT)
- •Radiographic testing (RT)
- •Ultrasonic testing (UT))
- •Eddy current testing (ET)
- •Thermal infrared testing (TIR)

Every NDE method has limitations and in most cases, a thorough examination will require an application of a minimum of two NDE methods[1]. Non-destructive evaluation can be used to ensure proper quality after the product manufacturing, or it can be used to continuously monitor some components. This is done to minimize the possibilities of failure, prevent disasters, and economically plan the replacement of components. NDE methods are commonly applied in the oil and gas industry, aeronautics, and various power plants including the nuclear power plant.

1.1 Ultrasonic testing

Ultrasonic testing can be used for the inspection of various materials such as metals and alloys, composites, ceramics, plastic, and sometimes even wood and concrete. There are several ways an ultrasonic testing technique can be implemented, but the main principles are always similar. Pulse-echo (PE) is one of the simple implementations of ultrasonic testing that consists of only

one transducer that serves both as the transmitter and the receiver of ultrasound energy. An illustration of PE is shown in Figure 1.1. An ultrasonic transducer is placed on top of some

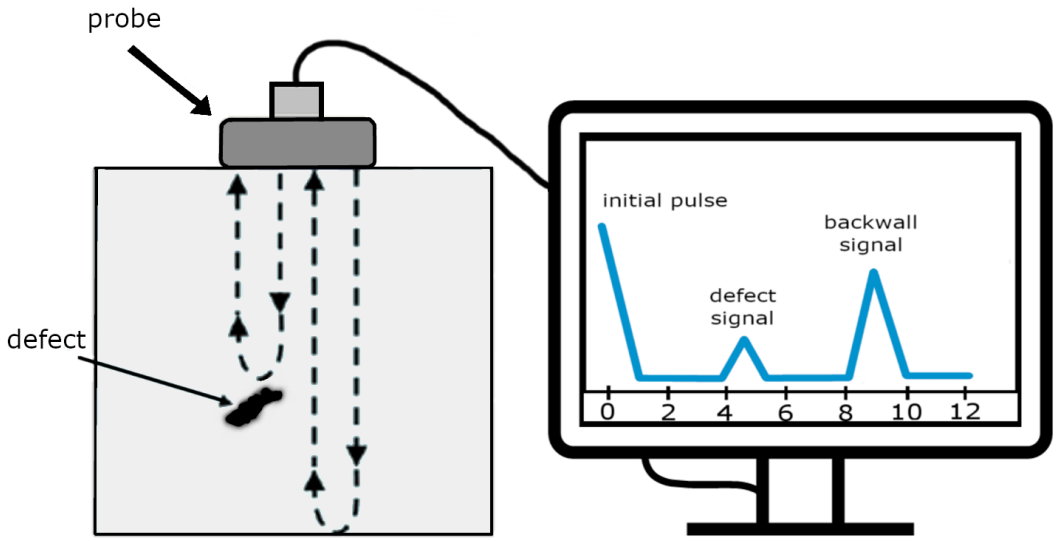


Figure 1.1: Illustration of the pulse-echo method for ultrasonic testing

material, and then it transmits ultrasound waves throughout the material. Whenever there is an inconsistency in material density, the acoustic impedance changes and causes the reflection of some of the ultrasonic waves. The first time this happens is when the waves are entering the material. A large portion of the energy will then be immediately reflected and when plotting the amount of received energy as a function of time (A-scan), this signal will appear at the beginning of the x-axis as shown on the plot displayed in Figure 1.1. The second time ultrasonic waves will reflect back to the probe is when the waves reach the bottom of the inspected component. This signal is called the backwall signal. When searching for a defect in the material, trained experts search for signals that appear between these two characteristic signals. If a signal is found in this area, it is probably caused by the defect that is positioned between the surface of the material and its bottom. The described procedure shows how the analysis of one A-scan is done. In reality, the inspection of the whole material is needed, so the probe must be moved along the surface. This is usually done with a robotic manipulator that consistently moves the probe from one side of the material to the other and collects a series of A-scans. The surface of the inspected material is often not perfectly smooth, so while moving the probe the air can appear in between the probe and the inspected material (lift-off). This causes a lot of noise in acquired data, so to prevent this from happening, the scanning can be performed with the probe and the material submerged in some liquid. Another possibility is to apply lubricant between the probe and the material, which prevents the air from getting in between the probe and the material. Once the probe was moved from one side of the material to the other, data from one cross-section of material was collected in the form of a series of A-scan. A series of A-scans can

also be converted into an ultrasonic image for easier manual analysis. This is done by converting each of the A-scans into one image column. Pixel intensities are determined from the amplitudes of corresponding A-scans. Also, since the x-axis of the A-scan represents the time needed for a signal to be reflected back to the probe, if the speed of ultrasonic waves thorough the inspected material is known, it is possible to calculate the exact depth of an artifact that caused some signal. This allows the inspector to directly report the coordinates of the defects found inside the material. PE technique is extensively used in industry due to its simplicity and efficiency [3]. However, it has one substantial disadvantage. If a flat defect is positioned parallel to the trajectory of transmitted ultrasonic waves, the surface from which the waves can possibly reflect is very small. This can easily lead to an undetected defect. To increase the reliability of finding a defect, scanning at various angles can be performed. However, if different probes are used to accomplish this, the time needed for data acquisition would be increased by several times. A better option is to use a single phased array probe. This probe has the ability to simultaneously scan the material from different angles. An illustration of a phased array probe is shown in 1.2. In Phased Array Ultrasonic Testing (PAUT), images are typically formed through constructive

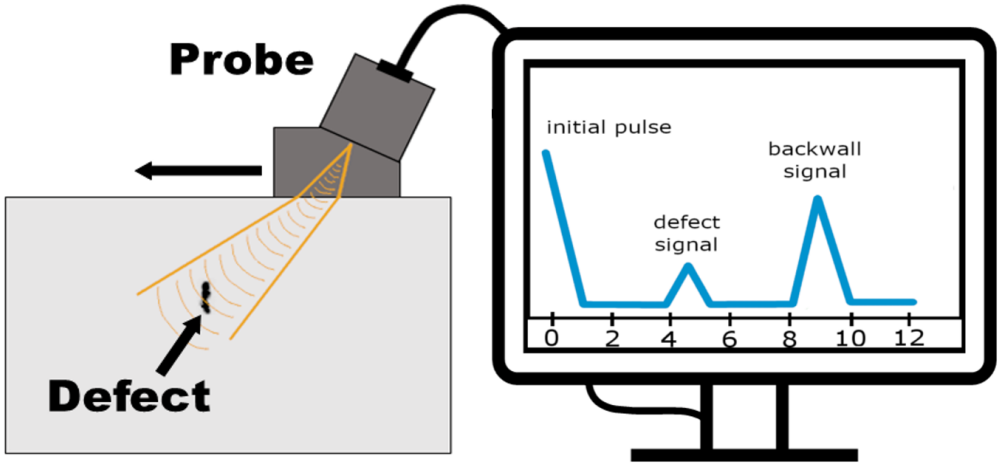


Figure 1.2: Illustration of the phased array ultrasonic testing (PAUT)

and destructive superposition of signals backscattered from flaws or geometric features [4]. The angle of the transmitted beam can be steered. This enables the phased array probe to collect data from dozens of angles in a single pass from one side of the material to the other. Some defects might not be visible from all angles, so all of the collected data must be analyzed. This increases the reliability of inspection, but it also means more data for inspectors to analyze.

Another ultrasonic testing implementation that was commonly used before is called Time Of Flight Diffraction (TOFD). Data obtained by this approach was not used in this thesis, so the principles of this approach will not be discussed here. In general, ultrasonic testing has many

advantages compared to the other Non-destructive testing techniques. Some of the important advantages are [5]:

- •detection of both surface and subsurface discontinuities
- •higher depth of penetration than other NDT methods
- •pulse-echo and phased array techniques require only single-sided access
- •highly accurate
- •requires minimal part preparation
- •instantaneous results

These advantages made ultrasonic testing a popular NDE approach for industry applications.

1.1.1 Ultrasonic testing data analysis

The amount of data collected during the ultrasonic inspection is immense, especially if a phased array probe is used and the cross-sections are acquired at various angles. The analysis of the UT data is currently done manually by trained personnel. This is done with the help of software that is able to process and display the acquired UT data in various formats. Such software has many functionalities that help the inspectors to analyze the data quicker and more confidently. However, commercial software for UT data analysis, currently do not contain advanced tools for automated UT data analysis and defect detection. During the ultrasonic data analysis, trained inspectors simultaneously look at various representations of data. Some commonly used representations are shown in Figure 1.3. The top left window contains an A-scan, which

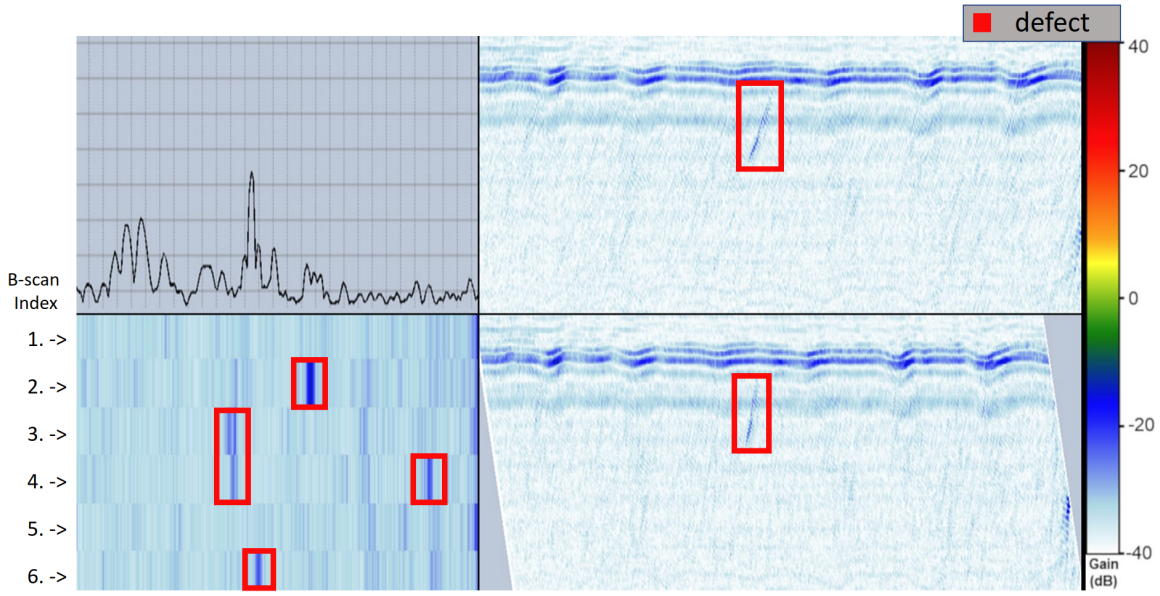


Figure 1.3: Various representations of UT data.

is a representation that displays the amount of received energy as a function of time. On the right side of the figure, two variants of B-scans are shown. The upper B-scan was obtained by

transferring each A-scan into one image column. The defects that are in reality vertical will appear slanted in this view because the angle of the ultrasonic waves is not taken into account. The lower variant is called volume corrected B-scan (VC-B-scan) and in this case, the A-scans are transferred onto an image at an angle that the ultrasonic waves were traveling through the material. VC-B-scans preserve realistic orientations of the artifacts found inside the material. In the bottom-left window, a C-scan representation is shown. This representation is obtained by projecting minimum values of some ultrasonic image columns, which means that each B-scan will be converted into one row. Values of each pixel of that row were calculated as the minimum value found in some columns of the B-scan. If this operation was done for the whole column of a B-scan, the projection would almost always return the minimum values caused by the initial pulse or the backwall signal, so the useful information would not be preserved. Instead of using all the pixels from a column, a C-scan projection is done only for the area between the initial pulse and the backwall signal. By looking at a C-scan, we can quickly get a rough idea about the B-scans that contain defects. By looking at Figure 1.3 it is clear that the B-scans with indices two, three, four, and six need a more detailed analysis. The inspection usually starts with that step. The inspectors then thoroughly check B-scans that looked suspicious on the C-scan. When analyzing some signal seen on a B-scan, the inspectors can also use other representations such as A-scan or sectorial scan (S-scan) to confirm their decision. Another thing that significantly helps when making a decision is to look at the surrounding B-scans. Many defects are not planar, so their signal will appear on multiple adjacent B-scans. Also, if the data is scanned with a phased array probe, it is useful to look at the same cross-section of the material from different angles. An example of these useful surrounding B-scans is illustrated in Figure 1.4.

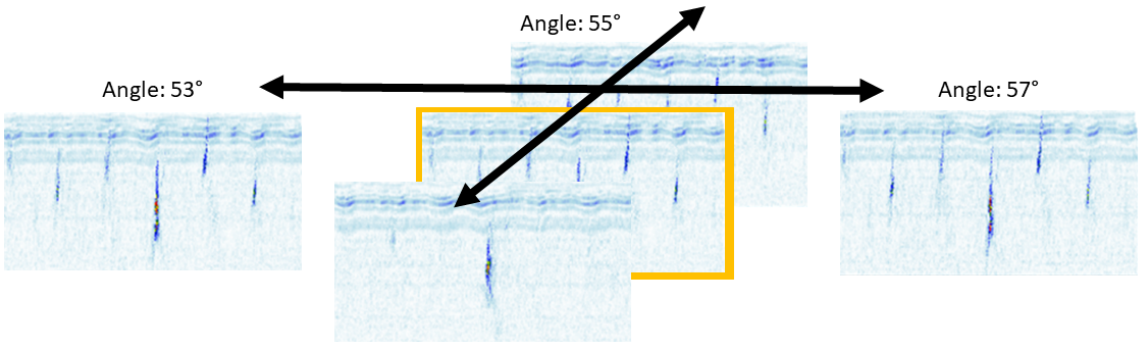


Figure 1.4: Sequences of ultrasonic B-scans. The same cross-section of material can be seen from multiple angles when the scanning is performed with a phased array probe. Neighboring cross-sections often contain the same defects and can be very useful during the analysis.

1.2 Problem description

There are several limitations in the procedure currently used to analyze UT data. First, there is a need for specially trained experts and their training requires a lot of time. Each inspection takes a significant amount of time, so the real-time results are usually not possible even if several inspectors are simultaneously working to analyze the data. Furthermore, decisions made by the inspectors can be subjective and prone to human errors, especially when a large amount of data needs to be processed which impacts the concentration of the inspectors. This problem will only become more prominent with the increasing usage of phased array probes that acquire more data compared to the traditional single-angle probes. Most of the collected UT data does not actually contain a defect. Having an algorithm that can extract only the suspicious parts of the data would significantly increase the analysis speed. Additionally, it would be very useful to precisely localize potential defects and pass all the suspicious signals to the inspectors for confirmation. This semi-automated approach is probably the intermediate step towards fully automated systems for ultrasonic testing data analysis.

In this work, a deep learning approach is used to analyze ultrasonic testing B-scans and to localize defects. Several problems must be tackled in order to create a precise and reliable algorithm for defect detection. A deep learning approach is chosen due to its superiority and better generalization compared to the traditional computer vision methods. However, in order for a deep learning approach to be reliable, a large enough dataset of images must be acquired for the training of such an approach. As stated before, the amount of data is increasing with the usage of phased array probes, but most of those images are empty, which is not ideal for training of supervised object detection model. An approach based on anomaly detection can be used in that case. Such approaches are trained solely on the normal data and are designed in a way that leads to higher anomaly scores when the model encounters data that differs significantly from the normal data used during the training. Another option that is used in this work is to collect the data by scanning components with artificially created defects inside the material. This way, a database that contains enough B-scans displaying defects is collected. Other problems that are encountered when applying object detectors for defect detection from ultrasonic images are the noise and the signals that appear due to reflections from the geometry of the scanned component. These signals can sometimes appear very similar to the defects' signals. A computer has no additional knowledge and input about the scanned component, which is something that human inspectors usually have access to when performing the analysis. This makes the decision-making process of some methods even harder. It is also one of the reasons for the poor performance of the traditional approaches, since the noise and geometry signals can hardly be anticipated and a method for defect detection must have a significant generalization ability while retaining reliability. There can also be some other challenges depending on the

types of components that are scanned (pipelines, bolts, solid metal blocks, welds, etc.) but such case-specific challenges are not discussed in detail here.

The focus of this thesis is the development of novel approaches based on deep learning for the analysis of ultrasonic B-scans. It is necessary that the proposed architectures can be trained with a small amount of data since blocks with artificial defects are expensive and in real-life the amount of images that can be collected for the training of a deep learning model is limited. As shown in the publications attached to this thesis, training on a small dataset can be accomplished by using transfer learning, extensive data augmentation, generation of artificial images, or by designing a simpler model with fewer parameters. Data augmentation and transfer learning are standard tricks when applying an existing object detector on a new dataset. However, artificial image generation with the goal of improving a detector's precision is not so trivial and thus not used as often as other data augmentation techniques. This topic is discussed and researched more thoroughly in Pub 3. Another problem that needs to be taken into account when developing a new object detection method is the extreme aspect ratios of the defects that need to be detected. In Chapter 3 the working principle of one-stage detectors is described. The influence of the anchors' design and placement on the loss function and the performance of the object detector is explained. The anchors' design and tweaks that are necessary to obtain the maximal performance out of object detector when analyzing ultrasonic B-scans are the topics of Pub 1 and Pub 2.

Another challenge with the current approaches for automated analysis of UT data is that they use only one cross-section of the material (for example one B-scan) during the decision-making. This approach is not ideal since useful information from the surrounding areas remains unused. Human inspectors always look at the suspicious areas of material from several viewpoints. Generally speaking, looking at the same material cross-section from different angles will produce similar images. The difference will be in the sizes of the defect's signal and usually, in the higher angles, the defects will appear more elongated. The only time a significant difference can occur is if the defect is planar and positioned in such a way that the ultrasonic waves do not reflect from it for a particular scanning angle. In that case, the defect will not be seen on some scanning angles, but it will probably appear on some others (that is the point of scanning the material with a phased array probe). This is why it is important to inspect data for all the scanning angles to ensure none of the defects will be missed. However, this slows down the inspection regardless of the way it is done (manual or automated). One of the contributions of this thesis is a novel method for simultaneous analysis of all scanning angles (Pub 4) that was designed to speed up the overall analysis without sacrificing reliability.

Simultaneous analysis of ultrasonic B-scans can also be used to increase the mean average precision of the defect detector. In this case, it is better to use the neighboring cross-sections of the material rather than the same cross-section as seen from a different angle. However, a simple

input expansion of the standard object detector will not enable the model to use efficiently this additional information. A more complex approach is needed, and designing such an architecture was the topic in Pub 5.

While there is a large number of approaches for image analysis, most of them are designed for some general computer vision tasks. Straightforward application of the existing method can sometimes lead to good results, but more often some tweaks are needed to achieve the desired performance. The existing methods can be notably improved if some NDE domain-specific knowledge is combined with the knowledge of computer vision and if the novel models are developed with this specific application in mind.

1.3 Scientific contributions

The emphasis of this thesis is on novel deep learning-based methods for the analysis of ultrasonic testing data. More specifically, one-stage object detectors are used to detect defects from ultrasonic B-scans. In order for this approach to work well and outperform existing methods, an architecture appropriate for the detection of objects with extreme aspect ratios must be developed. To further improve the results, a method for automated analysis of UT data must approach the current procedure for the analysis of UT data which relies on confirming decisions by looking at the same area from different viewpoints. This can be accomplished by expanding the input to the model and designing an architecture that can successfully capture this additional information and improve its decision-making process. The scientific contributions of this thesis that are the results of the performed research are the following:

- Method for detection of defects with extreme bounding box aspect ratios from ultrasound images based on deep one-stage detector.
- Method for defect detection by simultaneous analysis of multiple ultrasound images based on deep one-stage detector.

1.4 Thesis structure

The main contributions of the thesis are presented as a compilation of five research publications addressing the research objectives stated earlier. The thesis is structured as follows. Chapter 2 contains an overview of the existing methods for automated analysis of NDE data. Chapter 3 describes the existing approaches for object detection and their working principles. Chapter 4 contains definitions of commonly used metrics for evaluation of object detectors and methods for automated defect detections. The main scientific contributions of the thesis are presented in Chapter 5. Chapter 6 lists the publications used in this work, and in the following Chapter 7 the author's contribution to each individual publication is given. Finally, in Chapter 8 the

conclusion is written together with some possible future research directions.

Chapter 2

Overview of existing approaches for non-destructive evaluation data analysis

2.1 Methods for automated analysis of ultrasonic testing data

Ultrasonic testing can be used to inspect various materials. It is commonly used for inspection in aeronautics, the oil and gas industry, and all kinds of power plants. It is often applied to inspect metal components, but it can also be used to inspect carbon fiber reinforced polymer (CFRP) specimens, or some other materials such as ceramics, concrete, and wood. Since the inspection procedure differs depending on the inspected material, obtained data also differs, so a variety of methods was invented to automatically analyze collected data. Also, depending on the use case, the results of the analysis can be of different granulation. Most of the methods for automated analysis perform classification which means that for some samples such as one A-scan, or one B-scan it is possible to determine if it contains a defect's signal, but the exact location of the defect is not explicitly given by such algorithms. For A-scan analysis, this is not a problem, since one A-scan is already very localized and we can precisely determine the defect's real-world coordinates from that information. For B-scan, it is possible to perform a more thorough analysis and provide the location of the defect (object detection methods), or even a pixel-wise segmented map (semantic segmentation methods). If the algorithm for automated analysis is used in collaboration with a human inspector, it is useful if an algorithm can at least give a rough location of the defect so that the inspector knows which part of the data is considered anomalous. Also, a fine-grained localization enables automatic calculation of the dimensions of the defect, which can be used to assess the severity of a problem.

2.1.1 Methods for A-scan analysis

The defects can be recognized from the acquired UT signal because the reflections from material discontinuities appear in the A-scan as abrupt time localized changes resulting in time-varying spectral characteristics [6]. One of the popular approaches for automated defect detection from ultrasonic A-scans is to process the signal with wavelet transform and then feed the obtained coefficients into some classifier.

Signal processing with wavelet transform works by decomposing the signal into N levels and calculating appropriate approximation and detail coefficient. By thresholding the detail coefficients and calculating inverse transformation on the remaining data, it is possible to denoise the original signal. Usage of long time intervals to obtain more precise low-frequency information and shorter regions for obtaining high-frequency information is enabled by applying a windowing technique with variable-sized regions. Many authors used the wavelet transform to improve the quality of A-scans before performing further analysis. The authors of [7] built a classifier based on this procedure and used it for the classification of three distinct signals (fault echo, echo from weld, and backwall echo) in the material used for airplane engines. After processing the signals with the discrete wavelet transform, the authors calculated features such as mean value, standard deviation, etc. from the ultrasonic signal. The features were then classified by Support Vector Machine (SVM). A similar approach was used in [8] where the authors classified four different types of defects in stainless steel plates. The data consisted of 240 A-scans collected using a pulse-echo technique. Unlike [7], the authors of this work used an ANN to classify defects and achieved an average accuracy of 94%. There are also many other works [6, 9, 10] that applied similar approaches for the development of methods for automated classification of A-scans. These methods were developed for different types of UT data and sometimes use a slightly modified approach compared to the one described above. For example, in [9] the process described earlier was used for the analysis of ultrasonic TOFD data [9]. The authors concluded that the SVM classifier performs well even when the dataset is small, which is the main advantage compared to the ANN classifier. The authors of [6] used the envelope shape of the signal as an input to the ANN instead of calculating mean value, max value, or some other similar features usually calculated from the signal. Signal prepossessing with the wavelet can remain the same regardless of the used classifier and inputted features, but sometimes different features work better for different classifiers. In [11], a special type of ANN called Convolutional Neural Network (CNN) was used to analyze UT data obtained by scanning CFRP specimens. The authors additionally modified a standard CNN by swapping the last layer with an SVM, which improved the results. In order to feed the information extracted with wavelet transform to the network, calculated transformation coefficients were re-organized into a 2D matrix with dimensions 32x16, which was then used as an input. The authors concluded that this input works better than feeding hand-crated features from the coefficients. It

is also possible to use a different preprocessing / feature extraction technique such as DFT or Cosine transform [12] or to perform additional feature selection of the calculated features using a PCA [12, 13] or Wilcoxon-Mann-Whitney test [12]. Another option is to skip completely hand-crafted feature extraction and let a machine learning (ML) model automatically determine the important information from the data. The progress in deep neural networks and their applications to various domains has greatly stimulated research of such methods for the analysis of UT data [14, 15, 16, 17]. The authors of [14] collected TOFD and pulse-echo (PE) data. The author inputted A-scans into ANN, which was used to classify four different types of the signal (conditions of weld joints). In the case when the A-scans were first smoothed with a low-pass filter, the ANN achieved an accuracy of 73% for PE and 98% for TOFD signal classification. In [15] it was demonstrated that a deep neural network (DNN) with dropout regularization outperforms a simple ANN. The authors performed many experiments in the search for the best hyperparameters, such as the dimension of hidden layers and activation function choice. This work was also the first work to run experiments for automated analysis of UT data on a mixed frequency dataset. A DNN achieved significantly better accuracy than ANN on the task of classifying five defect types. The authors also noted that image representation of the data such as B-scans and C-scans are very useful in the context of non-destructive evaluation and that these representations should be considered in the future for the development of automated UT data analysis systems. In the follow-up work [16], the authors added Gaussian noise to collected A-scans and compared the performance of CNN and DNN for various levels of signal-to-noise ratio (SNR). The CNN network achieved on average 6.82% better accuracy compared to the DNN. The authors also noted the importance of data augmentation. The authors tested time-shifting of the defect signals, which mimics changing the distance between the transducer and defect in a real coordinate system. Data augmentation led to significant improvements for both the DNN and CNN, and for all the SNRs. In [17], a CNN in the form of an autoencoder was used to further improve the denoising abilities of CNN. Autoencoder is composed of three parts: encoder, latent layer, and decoder. The spatial resolution in the encoder is decreased by using the pooling layer. After the latent layer, the spatial information is increased by using upsampling layers. This bottleneck design forces the architecture to learn and extract important information from the input data. It was shown that such architecture works better with noisy data compared to the standard CNN. Similar to the previous work, data augmentation was used to increase the number of collected samples, and various levels of Gaussian noise were added to test the performances of the models. Autoencoder architecture was proven to work well for denoising and the performance was improved by several percents in different experiments by using this approach. By looking at the aforementioned related works, it can be seen that the recent trend is to collect a large enough database of A-scans and then used some direct approach without hand-crafted feature extraction. DNN and CNN proved to be especially suited for this

task. Also, many authors noted the importance of expanding their current work to work with ultrasonic images. However, the number of research work showing the usage of modern ML and DL-based solutions for automated analysis of UT data is limited, presumably because of the costs involved with the collection of a large enough dataset of ultrasonic images.

2.1.2 Methods for B-scan and C-scan analysis

Methods for automated UT data analysis have for a long time relied on approaches for A-scan analysis. Traditional well-investigated signal processing techniques could be applied to A-scan signals and the achieved results were good. On the other hand, traditional techniques for image analysis were not as good, so not many works went down this path. Despite this, there are some works that explore traditional approaches for automated analysis of UT images. Wavelet transform can also be a useful tool for feature extraction from images, as shown in [18]. The most relevant features were selected by PCA, and fed into a fuzzy C-mean clustering classifier. The proposed approaches were tested on several TOFD images with known geometric defects in them. Presented results show that the proposed approach can successfully segment different types of defects in an image. However, to ensure the reliability of mentioned method, it should be tested on a much larger and more diverse dataset of images. Another example of a traditional image processing technique for analysis of ultrasonic B-scans was shown in [19]. In this work, the authors used a Radon transform to detect cracks in rails. The data was obtained by scanning the rail with three different probes. Acquired B-scans were first processed with the wavelet decomposition, which was done to suppress the horizontal structures, thus eliminating the noise in the B-scans while preserving the defect's signal. The authors then used Radon transform to detect cracks in the denoised images. The authors concluded that in future work, a neural network approach should be built in order to create a fully automated system. In [20], TOFD B-scans were analyzed using a parabola matched filter. This is possible because the motion of the emitter and receiver relative to the scatterer such as defects describes the characteristic parabolic shape. One of the drawbacks of this approach is that the parabola's form varies with the depth of the defect, so this approach works only for a specific depth. The approach achieved good results when tested on simulated data, but on real experimental data, it was less effective. Traditional image processing techniques were also used for the analysis of ultrasonic C-scans [21]. In this work, the authors used a reference image, a C-scan, showing the flawless inspected component. If later a new instance of the same component is inspected, it is possible to compare the obtained C-scan with the expected one and highlight the differences. The authors reported no missed defects, but the number of false detection was very high. The robustness of this approach is probably affected by the fact that the data from different instances of the same component often looks different in real life.

Lately, a deep learning approach became a dominant approach for the analysis of sequences

and images since some architectures like Convolutional neural networks (CNN) have a natural ability to process sequences and grid-like representations of the data. For a long time, an approach based on deep learning has been hindered by the cost and difficulty of gathering enough data to train a good deep learning network [22]. If a big enough dataset of UT data is available, one can utilize additional context information available in the B-scan compared to the individual A-scan analysis, and develop a more precise method. This advantage was also noticed by many other authors, so the development of methods for B-scan analysis recently got a lot more attention. In [22] it was demonstrated how the data needed for the training of deep learning architecture can be simulated. The authors used simulated data to train a network for crack characterization. proposed deep learning approach was compared to the 6dB drop method. The deep learning model was able to size 97% of the tested defects of lengths 1 to 5 mm within ± 1 mm, while the 6dB method could only size 48% of the defects. In [4] the authors showed that a CNN trained mostly with simulated data in combination with a small amount of real data can be used to detect, locate and size a defect from ultrasonic phased array data. The used dataset was created by GPU-accelerated finite element simulations and then expanded with a small percentage of real data. The authors trained a two-stage detector Faster-RCNN [23] that reached the area under the curve of 0.95 when tested on simulated data. When testing the detector on real data with an intersection over union (IOU) threshold of 0.4, the model was able to locate 70% of the flat bottom hole defects. Another approach for artificially generating the UT images used for the development of an ML classification model was shown in [24]. The authors extracted signals of several defects and inserted them into ultrasonic B-scans that do not contain any flaws. Using this technique, 20000 images were generated and used to train a model similar to VGG[25]. The authors compared the probability of detection [26, 27] achieved by this model with the human performance and concluded that the ML approach works as well as human inspectors. However, using only the generated or simulated data for comparing the performances can give an unwanted advantage to the ML model. Since the artificially generated UT data do not contain all the variations that can appear in real situations, reliable evaluation should be performed on a large-enough dataset of real images [4, 28]. In [28] the authors trained a machine learning model on an artificially expanded dataset of multichannel phased array data of austenitic welds. A separate subset of realistic data was used to test the performance of an ML classification model and compare it to the human-level performance. The model almost managed to match the inspectors' performance, missing two out of nine flaws that were detected by the experts. In [29] another example of training a CNN for the classification of different types of defects from simulated images is shown. The authors reported an accuracy of over 90% for all types of tested defects. In [30] a real UT data was used for the training and the testing of deep learning networks for defect detection from ultrasonic B-scans. The authors tested SSD[31] and YOLOv3 [32] architectures and concluded that YOLOv3, which achieved a mean average

precision of 89.7 %, outperforms SSD. However, the testing dataset contained only 98 images. In [33] the authors proposed a deep learning approach for the identification of the geometrical elements of a weld. The proposed process allows the segmentation of 3D Phased-Array Ultrasonic testing scans. The segmentation of the welded joints does not give information about the quality of the inspected weld, but the geometrical information can be used to determine if the acquisition of the data was performed correctly. Calculated geometrical information can also enable algorithms for defect detection to position detected defects within the geometry of the weld. This increases the relevance of the UT analysis and provides more detailed overall results. The used dataset was created from 30 3D UT scans, and the model achieved a voxel accuracy of 96.76% and a dice score of 90.00% on the test subset. In [34] the authors collected ultrasonic data by inspecting additively manufactured specimens. The specimens' surfaces were on purpose created with a different level of roughness, which influences the signal-to-noise ratio. The goal of their work was to classify specimens into different categories according to their porosity content. To accomplish this, the author tested several architectures (CNN, DNN, MLP) for the classification of collected ultrasonic images and determined that the CNN model achieved the best result with an accuracy of 94.5%.

2.2 Methods for automated defect detection from other types of NDE data

The methods mentioned below do not make an exhaustive list of works from NDE domains other than UT. There are many other works from each of the below-mentioned domains, and the publications listed here are simply the examples used to show recent trends in applications similar to the one from this thesis.

2.2.1 Visual inspection

Visual inspection is a type of NDE technique used to inspect the surface of a material and detect abnormalities. It is applied in various industries and for inspection of different types of material such as concrete inspection [35], rail systems [36], products (wires[37], steel strips [38] and others), pipelines [39], wind turbine generators [40], etc. Some approaches for automated analysis of images collected during the visual inspections are shown in the rest of the section.

In [40] the authors showed that the features extracted with a pretrained deep learning convolutional model work better than the hand-crafted features. The goal of the work was to visually inspect the images of wind turbine blades and detect possible damages like cracks, paint peeling, etc. The authors compared the classification performance of the SVM model depending on the inputted features. They concluded that the features extracted with VGG[25] architecture lead to

better performance compared to hand-crafted descriptors such as Histogram of oriented gradients (HOG)[41] and Scale-invariant feature transform (SIFT) [42]. In [43] the authors created a convolutional neural network to inspect rolled steel strips. The goal was to classify images into seven categories depending on the defect that appears on the strip. The achieved results were very promising, especially if it is taken into consideration that many of the deep learning architectures and techniques used for improved training were not yet invented at the time of publication of this article. In [37] the authors develop a one-stage detector to analyze wire images and detect three different types of surface defects. The proposed model was created by enhancing the Darknet53 [32] backbone with attention module [44] and combining it with Feature Pyramid Network (FPN) [45]. Experiments showed that the proposed approach achieves a mean average precision (mAP) of 88.5% which is a significant improvement compared to the other similar methods that were used before for this task. The authors of [46] propose a method for railway shelling defect detection. The authors compared several deep learning classification models with traditional approaches based on hand-crafted features and SVM classifier. Deep learning convolutional networks, VGG and ResNet, achieved far superior results compared to approaches based on HOG, SIFT, and LBP.

2.2.2 Thermographic inspection

Thermographic inspection is one of the NDE techniques commonly used to inspect carbon fiber reinforced polymer/plastic (CFRP). This material is often used in aerospace industry, automotive industry, power plants (e.g. wind turbine blades), etc. so convenient methods for such data analysis are necessary.

In [47] the authors tested several architectures for automatic defect detection from thermographic inspection images. The best results were achieved by the architecture based on the pretrained VGG [25] on top of which a decoder part inspired by the U-net [48] architecture was added. The authors also tested an approach for temporal analysis of each pixel-value based on 3-layer LSTM [49]. However, this temporal model did not achieve good results for all the tested samples. In [50] the authors proposed a generative adversarial network (GAN)-based semantic segmentation model trained with a novel joint loss function. The authors develop this model with the goal of analyzing different types of data without adjusting the parameters of the model or requiring multiple models. They tested their approach on carbon fiber reinforced polymer/plastic (CFRP) specimens and compared the results with existing methods for semantic segmentation. The authors evaluated the model using the F-Score, and the model proposed in this work significantly outperformed other tested models. In [51] the authors used a pretrained Faster-RCNN object detector to detect defects from thermographic images. The model was trained using the images collected from the literature and validated on specimens produced using different sets of material in order to show the generalization ability of the proposed model.

The best among the tested variants of the model achieved a mean average precision (mAP) of 75 %.

2.2.3 Radiography inspection

One of the NDE techniques often used to inspect the internal structure of material besides the UT is radiography testing. This approach is commonly used to inspect metal parts in the automotive industry to ensure that none of the components contains some internal weaknesses that could lead to expensive failures in the future. This approach is more popular for testing independent components right after the manufacture, since later the process is more complicated and would require significant preparation and often disassembly of the system being inspected. In [52] the authors proposed a system for automated detection of defects from X-ray images using a deep learning object detector. They tested two popular architectures (FPN[45] and Faster-RCNN[23]). The used dataset proved to be very challenging, so the achieved mean average precision (mAP) was quite low. The authors concluded that the FPN is better suited for detecting small defects compared to the Faster R-CNN.

In [53] the authors dealt with common problems encountered when developing a method for automated defect detection from NDE data - a small dataset. The main idea proposed in this publication was to use a Wasserstein Generative Adversarial Network (WGAN) [54] for artificial dataset expansions. They tested their approach on two datasets, one of which is a dataset of X-ray images of welding joints. The authors tested Inception [55] and MobileNet [56] architectures and, in the end, created an ensemble with a bit larger weight put on the Inception since it slightly outperformed MobileNet. The final ensemble achieved accuracies of over 94% for all the classes.

In one of the earlier works [57], the authors designed a method for the detection of defects from X-ray images by analyzing the gray line curve of vertical weld scan lines. The method for detection relies on the existence of local minimum points in cases where a defect is present in the weld. After the potential defects are segmented with this method, features are extracted and fed into the SVM classifier. This step is used to remove false positives from the first step. Experimental testing showed that SVM reaches an almost perfect accuracy and surpasses Artificial Neural Network (ANN) and Fuzzy inference classifiers.

In [58] the authors trained a deep learning network in three stages and used it to detect air bubbles in engines. First, they train an autoencoder using normal images, which are easier to obtain. Only the encoding part from this network is used later to perform classification. Then, the weights of the encoder are frozen and the rest of the network (fully connected layers)is trained on both defective and normal images. In the last step, the whole network is fine-tuned jointly using both normal and defective images. The proposed approach yielded around 9% of false positives and around 6% of false negatives.

Chapter 3

Overview of deep learning-based object detection methods

Object detection is one of the fundamental tasks in the area of computer vision. The main goal of object detection methods is to determine if objects from particular categories (such as person, car, tree, etc.) are present in an image and what are the exact locations of those objects. Object detection can be applied in various domains such as autonomous driving and robotics, surveillance, agriculture, medicine, industrial inspection and manufacturing, sports, and many others. Since object detection is an old problem, many traditional approaches were developed to perform this task. However, for some years now, the deep learning paradigm has completely taken over this field. Deep learning models, usually based on convolutional neural networks (CNN) architectures, achieve great results both in real-life applications and many publicly available datasets for bench-marking novel object detection methods [59, 60, 61]. Convolutional neural networks were dominating in computer vision area ever since the authors of AlexNet[62] won the ILSVRC 2012 challenge [60]. The rise of CNNs was largely driven by the increasing availability of large-scale public datasets and the more accessible computing power.

3.1 Convolutional neural networks

The dominance of CNNs in computer vision was not a coincidence. Convolution operation can be regarded as a sliding window approach, a strategy that is intrinsic to visual processing, particularly when working with high-resolution images [63]. The first step in the development of the CNN model is the collection and annotation of the data for which the model will be used. The database can be considered as a set of examples x associated with target values (labels) y. The goal is to use an optimization technique to determine parameters θ of the model. Those parameters are directly used to map inputs to desired targets. The trained model should in theory

learn a function $f(\theta)$ that transforms the inputs to their corresponding targets $f(\theta) : X \mapsto Y$. In the case of classification, the desired target value is simply a number that represents some category. In the case of object detection, the targets contain five values for each object in the input image that needs to be detected. Those values represent the category to which some object belongs and the bounding box that encapsulates the object. Bounding box are usually expressed either as four corner points of the object $(x_{min}, y_{min}, x_{max}, y_{max})$ or the center point and the dimensions of the object $(x_c, y_c, width, height)$. The target values are used to define the loss function (objective function or cost function terms are also used). The loss function defines the distance between the values outputted by the network and the target values. In order to find optimal parameters of the model, the expected loss $J^*(\theta)$ over data generating distribution p_{data} must be minimized. In practice, this is not feasible, so the loss function is approximated from the collected training data:

$$J(\theta) = \frac{1}{N} \sum_{i=1}^{N} L(f(x_i, \theta), y_i) \tag{3.1}$$

One desirable property of the loss function is its differentiability. If a deep learning network is designed as a sequence of differentiable operations on the input data, one can propagate the error from the loss function through all layers of the network. Propagating error from the network's output to its input allows the network to update values of its internal parameters (θ) in a way that leads to more precise predictions. This process is also known as back-propagation and was first successfully used by LeCun for training the CNN network [64] on a task of handwritten digit recognition. Minimization of the loss function relies on first-order partial derivatives of the loss function with respect to the model parameters. As stated earlier, if all the operations in the network are differentiable, one can apply the chain rule and calculate the gradient $\nabla_\theta J(\theta)$ of the loss function with respect to the model parameters. Once the gradient is calculated, a method like stochastic gradient descend (SGD) can be applied to update the parameters of the model in each training step. Many other optimization methods were later built by upgrading upon the vanilla SGD. Some commonly used optimizers include RMSprop[65], ADAM [66], Adadelta [67] and Adagrad [68]. To aid the process of training and allow the network to focus on important information in the input, researchers have come up with many different layers and optimization procedures. The layers can be arranged and combined in numerous ways and researchers are constantly improving the layouts of existing CNNs which leads to the development of novel architectures, usually with improved performances. The improved performance can for example mean better accuracy, easier training, increased generalization ability, decreased inference speed, or something similar. The reason why CNN architectures work so well for images is the natural ability of such architectures to process sequences and grid-like representations of data. Convolutional layers operate on the input tensor by sliding a kernel over the input, multiplying the values of a kernel with the input at the current kernel position, and passing the

resulting values through some activation function such as ReLu, sigmoid, tanh, etc. The activation function is used to increase the capabilities of the network and allow the modeling of non-linear transformations. In practice, more than one kernel is usually used, so the output of a convolutional layer will have a depth equal to the number of used kernels. The spatial resolution of the output depends on the type of convolution. Padding of the input, kernel size, and stride are the factors that determine the spatial resolution of the output. Convolutional layers are often paired with some type of pooling layer, such as maximum pooling. Usage of this layer helps the network to focus on important information and reduces spatial resolution, which leads to a smaller number of parameters in the deeper layers. Recently, many deep learning models also used normalization layers, such as Batch Normalization [69]. By using the Batch Normalization, the training is faster and more stable, but those merits are only seen if a large enough batch size is used. Many state-of-the-art object detectors are based on CNN architectures. Variants and working principles of the most popular CNN-based object detectors are described in the following section.

3.2 Object detection architectures

Object detectors are often divided into two groups: (I) One-stage detectors and (II) Two-stage detectors. Most of the detectors from these two groups rely on the detection of objects from a dense grid of predefined rough guesses about the objects' position in the image. These rough guesses are called anchors, priors, or default boxes. This concept is explained in more detail in the rest of this section. However, there are also some different approaches for object detection that do not rely on the usage of anchors such as CornerNet [70], CenterNet [71], FCOS [72], and similar. While it is interesting to see a different idea for performing object detection, this type of detector can hardly compete with anchor-based detectors in terms of precision and time complexity trade-offs. Additionally, there has recently been an increase in the development of computer vision models that are based on transformers [73] instead of CNNs. Several such models were built specifically for object detection [74, 75] and achieved results comparable to commonly used one-stage and two-stage CNN-based detectors. In this work, the focus will be on the working principle of one-stage object detectors, since many object detectors of this type achieve state-of-the-art results [76, 77] and these models are often applied both in industry and research. An architecture of a CNN-based object detector is usually divided into three parts: (I) feature extractor (backbone) (II) neck (III) detection head. A feature extractor is used to extract important information from the image. This is done by applying some architecture with good classification abilities such as VGG [25], ResNet [78, 79], MobileNet [56, 80, 81], DenseNet [82], EfficientNet [83], or some other. Improvements made in the image classification task often have a direct impact on the results for the object detection task. A better classification model

usually extracts better features and can be used to build more precise object detectors. Fully connected layers from the classifiers are discarded when reusing the classification model for the object detection backbone. Instead, the layers from the classifier's feature extractor are passed to the second building block of an object detector - the neck. The neck is used to combine high-resolution feature maps and low-resolution feature maps. Spatial resolution is usually decreased through the model by using maximum or minimum pooling layers or by performing convolutions with strides larger than one. While lowering spatial resolution means the fine details are lost, the model is able to focus more on the semantic meaning of the inputted information. One of the popular neck implementations is called Feature Pyramid Network (FPN) [45]. This idea was used as a baseline for the development of many other approaches for combining feature maps of different resolutions, such as PANet [84] and BiFPN [76]. Combining fine-grained details from earlier feature maps with semantic information from deeper layers improves the network's performance and allows easier and more natural image analysis on different scales. An illustration showing an object detector's building blocks is shown in Figure 3.1. After the

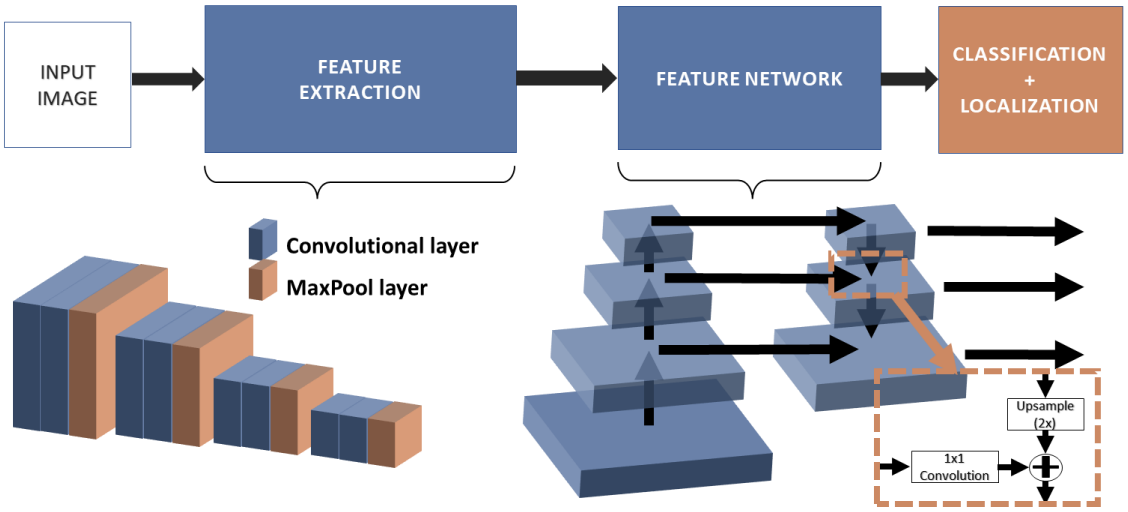


Figure 3.1: An illustration of object detector's main components.

multiscale features were extracted from the image using the backbone and neck of the model, they need to be fed into a detection head that will perform the actual classification and localization of the objects. A detection head can perform object classification and localization in one step (one-stage detectors) or two steps (two-stage detectors). With two-stage detectors such as the Faster-RCNN family [23, 85, 86, 87, 88], rough locations of the objects are first estimated. This estimation is done by the Region Proposal Network (RPN) which relies on features extracted by the backbone, and a dense grid of initial rough guesses about the possible regions of interest called anchors. Anchors are defined and placed in a way that covers the image with tens of thousands of bounding boxes with varying aspect ratios and scales. This dense grid of anchors is, together with the feature maps outputted by the neck, fed into the detection head as

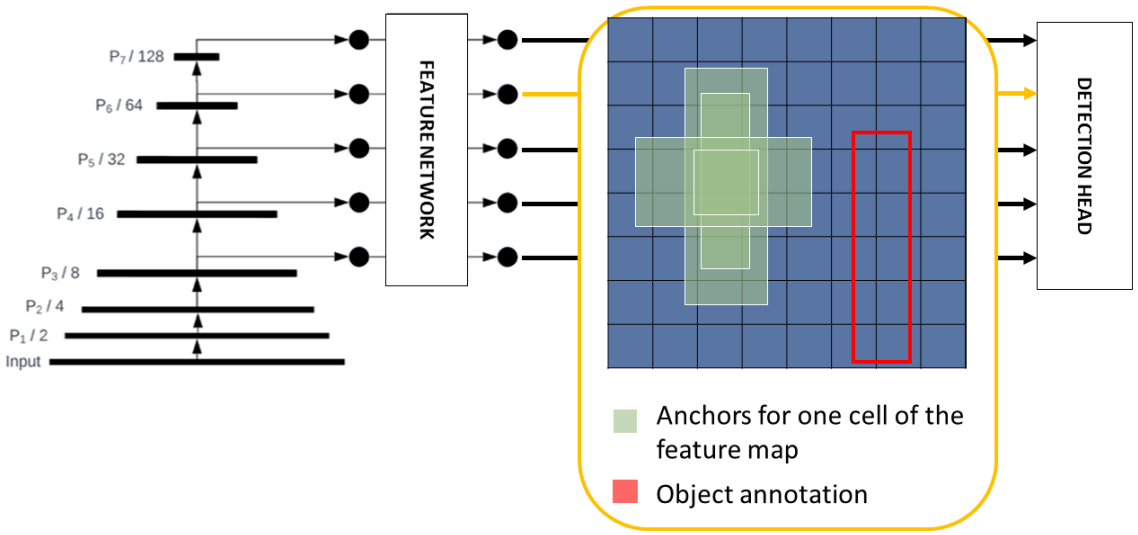


Figure 3.2: An illustration of the data inputted to the detection head.

shown in Figure 3.2. Anchors that probably contain some object are separated by the RPN and the shapes and locations of the selected anchors are refined by the network to obtain regions of interest. These are just the class-agnostic areas of an image that have a higher probability of containing an object. Regions of interest are then fed to a Region of Interest Pooling (ROI Pooling) layer together with the feature maps from the backbone. ROI Pooling divides the region of interest into smaller sub-windows and performs a pooling operation in each of these sub-windows. This layer is a special type of Spatial Pyramid Pooling (SPP) [89] layer, so the outputted features are of fixed size regardless of the input size. Finally, the calculated features are fed into branches for classification and regression. These branches will refine the regions of interest into the final predictions by discarding the regions without the object and fine-tuning the locations of regions that contain an object. The classification branch determines the category to which the detected objects belong. The described working principle corresponds to the way the Faster-RCNN works. Some other two-stage detectors have a different architecture and use slightly different approaches to perform object detection. The two-stage approach has additional computational complexity compared to the one-stage detectors. Despite that, it was for a long time a preferred approach in cases where accuracy was the deciding factor. Lately, some problems that were present in the early one-stage object detectors were solved and this allowed the one-stage object detectors to achieve state-of-the-art results [76, 77].

One stage detectors directly classify and localize objects in an image. This is again done by having a dense grid of anchors (also called priors, or default boxes). The principle is the same as the one used by the two-stage detector's RPN. The main difference is that for one-stage detectors there is no additional refinement, so the detection head must directly produce the final prediction from the anchors. This includes both the classification and localization tasks. Anchors' shapes and sizes as well as their placement on top of the image are determined from the

hyperparameters. This can have a huge impact on the model's performance. As seen in Equation 3.2 both the localization and the classification losses used during the training depend on the anchors that are selected during the training. The development of one-stage object detectors took off with the introduction of SSD [31] and YOLO [90] architectures. Many state-of-the-art models were later built by upgrading upon these architectures. Some examples that were created by upgrading SSD are DSSD [91] RetinaNet [92], RefineDet [93], and EfficientDet [76]. Examples of architecture that derive from YOLO include YOLOv2 [94], YOLOv3 [32], YOLOv4 [77], and YOLOv5 [95]. The improved versions of object detectors usually introduce ideas from other related computer vision tasks, such as image classification. For example, introducing layers from image classification models such as Batch Normalization or using some novel data augmentation techniques. Besides applying existing ideas to object detection, some publications also focus on redesigning the components of object detectors such as architecture's neck, which was improved in EfficientDet. Despite the differences among these object detectors, their working principle is similar. One-stage detectors all rely on predicting the locations and classes from anchors. On a high level, this can be considered as a modernized and better-optimized version of a traditional sliding window approach. For each window (anchor), the model needs to predict whether it contains an object and if it does to which category it belongs. Furthermore, the locations of objects are predicted as offsets relative to the corresponding anchors. The loss function used in SSD architecture [31] is shown below to explain how the optimization process is used to train one-stage object detectors. The overall loss function of a one-stage object detector is usually expressed as a weighted sum of localization and classification loss functions:

$$L_{loc}(x,l,g) = \frac{1}{N}(L_{conf}(x,c) + \alpha L_{loc}(x,l,g)) \tag{3.2}$$

where: $x_{i,j}^{p}$ = indicator for matching the i-th anchor box to the j-th ground truth box of class p

l = predicted bounding box

g = ground truth bounding box

α = weight of localization error

An example of the classification loss function is the cross-entropy loss:

$$L_{conf}(x,c) = -\sum_{i \in Pos}^{N} x_{i,j}^{p} \log(\hat{c}_i^p) - \sum_{i \in Neg}^{N} \log(\hat{c}_i^0) \tag{3.3a}$$

$$\hat{c}_i^p = \frac{exp(c_i^p)}{\sum_p exp(c_i^p)} \tag{3.3b}$$

where: $x_{i,j}^{p}$ = indicator for matching the i-th anchor box to the j-th ground truth box of class p

c = predicted class confidence

Localization error:

$$L_{loc}(x,l,g) = \sum_{i \in Pos}^{N} \sum_{m \in \{cx,cy,w,h\}} x_{i,j}^{p} smooth_{L1}(l_i^m - \hat{g}_j^m) \tag{3.4a}$$

$$\hat{g}_j^{cx} = (g_j^c x - d_i^c x)/d_i^w \tag{3.4b}$$

$$\hat{g}_j^{cy} = (g_j^c y - d_i^c y)/d_i^h \tag{3.4c}$$

$$\hat{g}_j^{w} = \log\left(\frac{g_j^w}{d_i^w}\right) \tag{3.4d}$$

$$\hat{g}_j^{h} = \log\left(\frac{g_j^h}{d_i^h}\right) \tag{3.4e}$$

where: $x_{i,j}^{p}$ = indicator for matching the i-th anchor box to the j-th ground truth box of class p

$smooth_L1$ = smoothed L1 error

l = predicted bounding box

g = ground truth bounding box

d = anchor box

cx, cy = center coordinates of a bounding box

w, h = width and height of a bounding box

As it can be seen from the equations, both the localization and classification losses depend on chosen positive anchors. The chosen anchors are the ones for which the variable x has value one. Usually, the positive anchors are the ones that overlap with the ground truth label by more than some threshold. This overlap is calculated as intersection over union metric (Jaccard index). Additionally, if some ground truth label does not have an anchor that overlaps by more than the defined threshold, an anchor that fits the best to the ground truth is used (even if its overlap is smaller than the threshold). Proper anchor shapes lead to more sampled anchors and better initial guesses of the object appearances, and are thus very important hyperparameters. In [94] the authors proposed the usage of K-means clustering on training annotations in order

to compute good anchor shapes. There were also attempts to optimize anchors for other object detectors [96, 97, 98]. Optimization of anchors is especially important in cases where the objects are extremely elongated. If the objects have an extreme aspect ratio, default anchors settings will not produce enough matched anchors for training and will thus negatively impact the detector's performance. Extreme aspect ratios are common in ultrasonic images due to the angle of data acquisition. When the probe transmits ultrasonic waves at large angles (for example 60°-80°) the defects and other structures will appear elongated in the resulting image and can have an aspect ratio greater than 10. This is why a significant portion of the work in this thesis was dedicated to the development of proper anchors design, placement, and matching.

Another option, that would implicitly solve this problem, is the generation of additional training data. If due to imperfect anchor hyperparameters there aren't enough sampled positive samples for training, the problem may partially be mitigated by generating more training images. This would in turn increase the total number of anchors that are used during the model training. Additional training images can be generated using different approaches, but the usage of Generative Adversarial Network (GAN) - based methods stands out due to their ability to generate highly-realistic images. Generating additional training examples also helps the model training when the dataset is small, since the model sees more variations during the training. The described idea was used in multiple works [99, 100] to improve the object detector's performance.

3.3 Object detection from sequences of images

Another goal of this work was the development of a method that can detect a defect on some B-scan while using additional information from other images. The ultrasonic images can be expanded to sequences which would increase the amount of information inputted into a model. The sequences of ultrasonic B-scans can be formed in two ways: (I) stacking images acquired for different scanning angles (horizontal axis in Figure 3.3) or (II) by stacking images in the scanning direction (vertical axis in Figure 3.3). If the first option is used to create sequences, object detection is somewhat similar to object detection in videos. The similarity stems from the fact that the neighboring frames look alike, and some distant frames can influence predictions made for some other frame. Modeling a long-term dependency between the frames can thus be beneficial. The second described way of creating a sequence is not so similar to the video sequence. The neighboring B-scans can be substantially different, and the important information will always be contained only in the space close to the target B-scan. Modeling a long-term relationship between the B-scans of the sequence is unlikely to yield any improvement in detection. This use case is more similar to the analysis of medical images than the analysis of videos. A thorough search of the relevant literature did not yield any research ar-

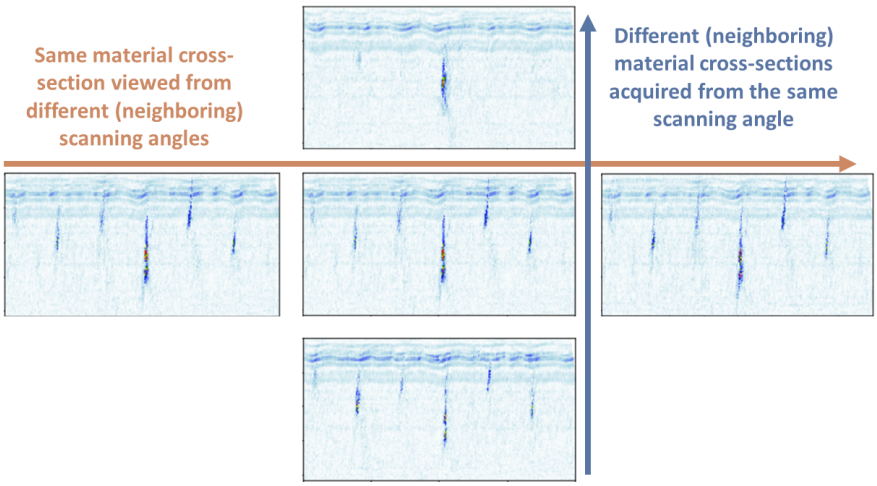


Figure 3.3: When analyzing ultrasonic B-scan, additional context can be obtained by looking at the same material cross-section from a different angle, or by looking at the neighboring material cross-sections.

ticles related to the analysis of sequences of ultrasonic testing B-scans. This is not surprising since most of the researchers do not have a large enough dataset to even train a deep learning model for image analysis, and expansion to 3D would require even more data. The inspiration for the development of methods for that task can still be found in some other domains and tasks that are somewhat related to this topic. As mentioned before, video analysis and medical data analysis are two tasks that share some properties with the defect detection from sequences of UT B-scans. An overview of methods from these areas is given below.

Methods for object detection in videos: When detecting an object from sequences of images, several approaches can be used. One simple approach is to post-process independent detection from each frame with an algorithm such as SeqNMS [101]. SeqNMS uses high-scoring object detections to increase the scores of related weaker detections from the nearby frames within the same video. Other options try to combine features from different frames to improve the precision. In [102] the authors use a flow-based approach to aggregate features from different frames. The authors designed a network that estimates the flow field and uses it to propagate features calculated from sparse key frames to other frames. This is much faster than calculating features for each frame by CNN. Another option that is often seen is the usage of 3D convolutional layers [103, 104]. The authors of [103] proposed a new architecture, called I3D, that was developed to take advantage of pretraining the model on a large-scale dataset as it is commonly done for image classification. Their model is built upon standard image classification architectures but with filters and pooling kernels inflated into 3D. Later works (S3D) [104] showed that it is possible to replace many of the 3D convolutions with low-cost 2D convolutions. The authors concluded that the 3D convolutions are more useful at the end of the network, where they enable temporal modeling between high-level semantic features. This also has the additional benefit of making the network faster compared to the version that uses 3D

convolutions at the beginning of the network. In [105] the authors propose a method that learns to index into a long-term memory bank while performing object detection. The authors augment the Faster-RCNN architecture by adding attention-based modules before the detection head. This enables the model to incorporate features outputted by the region proposal network (RPN) with the ones from the "memory bank". The combined features are then used to detect objects in the current frame. The authors of [106] propose a Spatio-temporal Sampling Network (STSN) that performs object detection on the current (reference) frame by using features calculated from some other (supporting) frame. First, a CNN computes object-level features for each video frame individually. Then, spatio-temporal sampling blocks are applied to the object-level feature maps in order to sample relevant features from nearby frames. This part is done by predicting a location offset from the combination of reference and support frame (target frame and context), and then extracting the features from the supporting frame with a deformable convolution. The sampled features are then aggregated into a single tensor, which is used as an input to the detection network to produce final object detection results for the given reference frame.

Ideas from the aforementioned works related to video analysis can be used as inspiration when designing a method for UT B-scan sequence analysis. However, there are many differences between these two tasks, so the direct application of some method for video analysis might not work that well for defect detection from sequences of UT B-scans. In video, there is a temporal dimension with a strictly defined orientation of increment. Most of the methods for object detection from the video are designed to work in real-time, which means that the future frames can not be used for the analysis of the current frame. This is not the case for either of the described ultrasonic sequences, since all of the B-scans can always be used to increase the amount of inputted information. Also, the main problems encountered in object detection in the video such as motion blur, video defocus, unusual poses, or object occlusions [106] do not appear when analyzing sequences of UT images.

Methods for object detection in Medical data: A domain that is more similar to the one investigated in this work is object detection from medical images. Unnatural images, small datasets, irregular objects that are difficult to distinguish from the background are some common challenges found both in medical data analysis and UT data analysis. The authors of [107] combined U-net and RetinaNet models to combine object detection with auxiliary semantic segmentation task. The developed architecture was used for medical object detection from CT and MRI data. The authors showed that additional training signals from the pixel-wise annotations can successfully be used to improve the results. Tested object detectors were implemented to work with both 2D and 3D input data. In [108] The authors designed a cascade framework that first proposes regions or volumes of interest and then uses a CNN to classify all the candidates. The first part is designed in a way that maximizes sensitivity with the cost of higher

false-positive calls. The false positives are then filtered by the CNN that analyzes features aggregated from randomly sampled sets of 2D or 2.5D views. The authors test their approach on several medical datasets (sclerotic metastasis detection, lymph node detection, and colonic polyp detection). A similar approach was later used in [109]. However, the approach introduced in this work is fully three-dimensional. The candidate regions are selected by a U-net-inspired two-stage detector Faster-R-CNN that was modified to work in 3D. The false positives are then reduced with a 3D DCNN. In [110] the authors propose a network that is able to incorporate 3D context when analyzing CT scans. Multiple neighboring slices are sent into a 2D detection network to generate feature maps separately, which are then aggregated for final prediction. The authors used R-FCN [87] as a starting point, and then make modifications necessary for it to work with 3D context. This approach is similar to the approach used in Pub 5. In [111] the authors develop a YOLO-based 2.5D fusion algorithm to localize individual 3D cells in densely packed volumes. Their approach is based on the fusion of 2D detections from orthogonal planes in 3D, which is then used to estimate the coordinates of the 3D bounding box. A similar approach was shown in [112] for the analysis of CT data. The authors propose a method that localizes anatomical structures in 3D images by first determining their presence in 2D image slices. In [113], the authors propose an optimized version of the SSD model for liver lesions detection from multiphase CT data. The goal is to design a model that can use knowledge from all the phases individually. This can not be accomplished by using standard convolutional layers, since the data distribution from each of the input phases is different. Instead, the authors applied convolution with separate filters for each phase and then concatenate the outputs into the resulting feature map. The authors then inserted an additional 1x1 convolution before the detection head to fuse the information from different phases.

While the overview of detection methods from medical data given here is not exhaustive, it is possible to get an idea about the main research directions. Most of the methods rely on feature aggregation extracted from 2D views and usage of additional context to improve the detection. The reason why this is the proffered way has mostly to do with the re-usability of existing architectures for image classification and lack of data to properly train full 3D convolutional networks. None of the mentioned methods for video analysis or medical data analysis were directly used to analyze UT data, since there are some differences between these domains. However, research works mentioned in this chapter served as an inspiration when designing a network for defect detection from sequences of UT data.

Chapter 4

Evaluating the performance of defect detection methods

4.1 Ultrasonic testing dataset

To develop novel methods for the automated analysis of phased array ultrasonic testing (PAUT) data, and to evaluate the performance of those methods, a dataset of such images must be available. Unfortunately, there are currently no publicly available datasets of realistic PAUT data. The only publicly available datasets are either artificially generated [24] or acquired with a different ultrasonic testing setup [114, 115] instead of a phased array probe. We collected the largest dataset of PAUT B-scans that was so far used in the literature in order to develop and evaluate methods for automated analysis of ultrasonic images. A large dataset enables the application of deep learning-based approaches and ensures the credibility of the achieved results. The dataset was obtained by scanning several steel blocks with artificially placed defects inside of them. The blocks were scanned with a phased array probe using the angles from 45° - 79° with a two-degree increment. The blocks contained a total of 68 defects, and most of them could be seen from various angles and scanning directions. This means that the same defect appears on several B-scans, and its appearance slightly varies in each of those scans. More than 4000 ultrasonic B-scans were collected and defects in those images were manually annotated. More details about the dataset can be found in the publications attached to this thesis (for example Pub 1). The methods proposed in this thesis were all developed and tested on the same dataset in order to allow comparability among the used approaches. Our experiments showed that the object detectors work a bit better when the pseudo-colored images are used as an input instead of unprocessed grayscale images. This is why we used pseudo-colored images in all publications except in Pub 3. Pub 3 uses a generative adversarial network to expand the dataset of images for training and since the generation of grayscale images is easier, in that work the original grayscale images were used for the experiments.

4.2 Evaluation metrics

4.2.1 Accuracy, precision, recall

The evaluation metric used to numerically define the performance of the model depends on the granularity of the algorithm's output and the goals of the evaluation. In cases where an image classification method is applied to analyze ultrasonic images, it is natural to also adopt commonly used metrics for the evaluation of image classifiers. For example, if an image classifier is applied to determine whether an ultrasonic B-scan has a defect or not, the accuracy metric can be used to quantify performance. It is important that the dataset is balanced if this metric is used. Accuracy is defined as:

$$accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{4.1}$$

where: TP = Number of true positives

TN = Number of true negatives

FP = Number of false positives

FN = Number of false negatives

Positive class is usually defined as a class of interest, so in this case, images containing defects would be positive examples. Correctly classified positive images are considered true positives. Images containing defects that were classified as normal images are called false negatives. Images that do not contain defects are negative examples. Correctly classified such images are called true negatives, and incorrectly classified such examples are false positives. When taking into consideration the domain where the methods for automated defect detection are applied, one can conclude that false negatives are much more serious than false positives. This is why a different metric such as precision and recall might be more suitable. The precision metric is similar to accuracy, but focuses only on the positive samples. It measures the percentage of true positives among all the examples that were classified as positive examples:

$$precision = \frac{TP}{TP + FP} \tag{4.2}$$

Another important aspect of some method's performance is to numerically define how many of the examples belonging to the important (positive) class were successfully classified. This is what a recall metric is used for. The recall is determined by the percentage of the positive

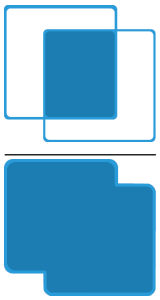
examples that some method was able to find from all of the positive examples:

$$recall = \frac{TP}{TP + FN} \tag{4.3}$$

4.2.2 Mean average precision

Optimizing independently one of the aforementioned metrics is easier than optimizing their combination. However, a method that has a 100 % recall is useless if the models simply always predict the positive class. This is why, in practice, we want to measure some overall metric that captures several of the aforementioned metrics. A metric that considers both precision and recall can be calculated as the area under the precision-recall curve (PR curve). When generalizing the described metric for the task of object detection, some additional aspects must be considered. First, the definition of true positives, false positives, true negatives, and false negatives must be slightly altered since the bounding boxes must also be taken into account. In order to define a correct prediction, the overlap between the predicted bounding boxes and the ground truth boxes must be quantified. This is usually done by calculating the intersection over union (IOU):

$$IOU = \tag{4.4}$$



If the IOU is larger than some threshold and the class of the predicted object is correct, the prediction is considered a true positive. If the predicted bounding box does not overlap enough with the ground truth bounding box or the predicted class does not match the actual class, the prediction is considered a false positive. Ground truth bounding boxes that do not have matched predictions are regarded as false negatives. True negatives would be all the other possible bounding boxes that can be found in the image that do not overlap with the annotated objects. There can be an infinite number of such boxes, so this value is usually discarded when analyzing the object detection results. Alternatively, one can consider all the anchors that were correctly classified as the background class to be the true negatives. In practice, this would mean that tens of thousands of correctly classified negative anchors are considered as true negatives (if an object detector works well). Since accuracy depends on the number of true negatives, it is not an appropriate metric for evaluating object detection methods. Object detection is not limited to the detection of only one type of object. It is much more common that the object detector must be able to distinguish between multiple objects and to be able to localize all of them in the image. To calculate the performance of an object detector for this task, the previously

mentioned area under the PR curve metric can be expanded to work with multiple classes. A mean average precision metric is used for this. It is calculated in the following way:

1. Sort prediction of a model by the confidence and assign them to the matching ground truth
2. Each prediction that has an IOU greater than some threshold (usually 0.5) and has a correct class is matched to the ground truth.
3. The prediction is correct (true positive) only if the ground truth was not already assigned to some other prediction. Otherwise, the prediction is considered a false positive.

The result of the described procedure is a list of predictions, both correct and wrong, that are sorted by their confidence. A precision-recall curve can be plotted by gradually taking examples from the list. The recall will either increase or stay the same as more examples are taken from the list. The precision can either increase or decrease, and usually the precision-recall curve will have a zigzag shape depending on the number of TP and FP that are found in each of the sampled lists. To reduce the influence of these small variations in precision, the PR curve needs to be smoothed before calculating the average precision. This is done by replacing each of the precision values with the maximum precision value to the right of the current value (future precision values obtained for higher recall values). Illustrations of PR curves before and after the smoothing is shown in Figure 4.1. The average precision (AP) is defined as the area under the PR curve:

$$AP = \int_0^1 p(r)dr \tag{4.5}$$

where: r = recall

$p(r)$ = smoothed precision-recall curve

The smoothed curve is more commonly used for the calculation of AP and in that case, the equation can be written in another way. The average precision can be calculated by summing the areas of rectangle surfaces underneath the smoothed curve formed by sampling the curve in points where the maximal precision was decreased:

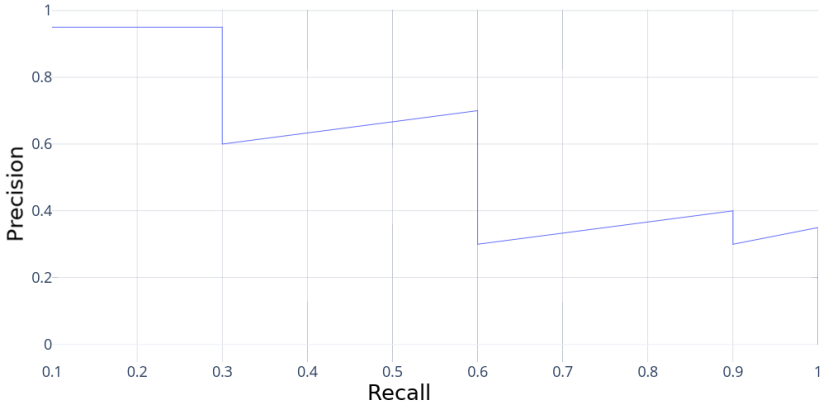
$$AP = \sum_{k \in K} (r_{k+1} - r_k) p_{inter}(r_{k+1}) \tag{4.6}$$

where: r = recall

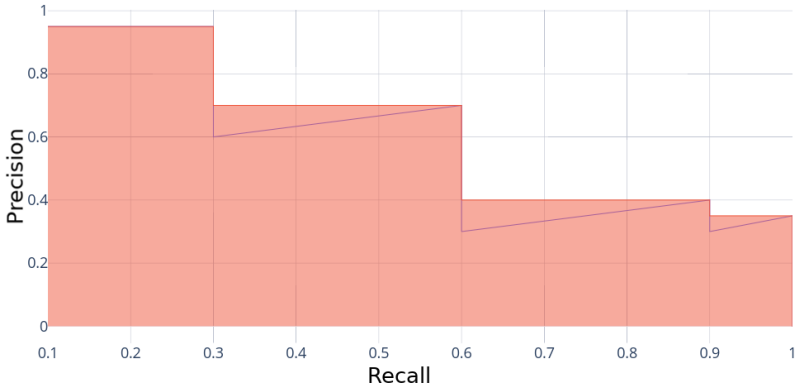
p_{inter} = smoothed precision-recall curve

K = list of indices for which the decrease of maximal precision occurred

Once the average precision is calculated for each class, a mean average precision (MAP)



(a) before smoothing



(b) after smoothing

Figure 4.1: Precision recall curve

can be calculated. This metric is obtained by calculating the average performance across all the classes. This metric is the most common metric for the comparison of object detectors on popular public challenges such as PASCAL VOC[61], COCO[59], and ImageNet[60]. Since defect detection in B-scans is an equivalent task to standard object detection, MAP is suitable to determine the performance of defect detectors.

4.2.3 Probability of detection

Since the topic of this thesis is the application of object detection methods for defect detection of ultrasonic testing images, it is important to mention another evaluation approach called the probability of detection (POD) [26, 27, 116]. This metric is more tightly related to the NDE domain, and it is often used to determine the overall reliability of the whole inspection procedure. This includes the hardware for acquisition as well as the analysis of the data done by the human inspectors. However, a unique way of POD approach does not exist, instead, a number of evaluation methods that are distinguished for example by the signal inputs or used statistical methods

[116] are used in practice. The drawback of the previously described MAP metric is that it does not take into account the defect sizes. A defect's size is an important factor, since the severity of the problem that can be caused by the material failure depends primarily on the dimensions of the found defect. When a method for automated defect detection is used, it analyzes all the angles, so a defect's size in pixels can vary a lot. A defect with a smaller physical size can sometimes appear bigger on a B-scan compared to a defect with a smaller physical size. Also, if the test dataset is small, it is possible to achieve great results even though the actual performance on new cases might significantly vary. For example, if only one defect is contained in the test set, a model can achieve a perfect score but the model's performance on new cases is highly uncertain. Probability of detection (POD) is a tool based on the advanced statistical analysis of the obtained detection results, used to calculate the reliability of inspection procedures with sufficient certainty. An example of this metric is hit/miss POD. It is calculated from the list of flaws, their sizes, and hit/miss label indicating whether a flaw was successfully detected or not. This type of POD curve is plotted by placing the flaw sizes on the x-axis and the corresponding probability of detections on the y-axis. A logit/probit curve can then be fitted to the data and the interval for a specific confidence level can be found. A point from this curve will determine the smallest flaw a procedure can reliably detect. A commonly used threshold is $a_{90/95}$ [24, 28, 117]. This means that the probability of detection must be over 90% with the confidence of the obtained results of at least 95%. An example of such a curve is shown in Figure 4.2. The plot shows that the smallest flaw that can reliably be found is 2.1 mm because this value on the x-axis is obtained for the y-axis value corresponding to a 90% probability of detection in the lower bound of the 95% confidence curve. The main drawback to this evaluation metric is the large number of flaws that are needed to perform the analysis. Also, the threshold discussion and complex relationships between the NDE response and the defect might make it impossible to use POD analysis in a regular way [116]. If the hit/miss POD analysis is performed for an image classification method that analyzes individual B-scans as done in some previous works [24, 28] a criterion for hit/miss is straightforward. If the model correctly classifies a B-scan with a defect, the prediction is regarded as a hit, if the model falsely classifies such B-scans it would be considered a miss. However, in our use case where the PAUT data is analyzed with object detectors, there are many additional criteria that need to be determined. How much does the predicted bounding box need to overlap with the ground truth to mark a detection as a hit? How to ensure that the ground truth bounding boxes are completely correct and would not be annotated differently by another inspector? If a defect appears on multiple B-scans, how many of its appearances need to be detected for it to be considered a hit? What to do with borderline cases where the defect's signal is barely visible, and a human inspector would also not be able to make a decision without looking at additional data? These questions make the usage of POD very challenging, and they ultimately prevented the usage of POD analysis in the publications

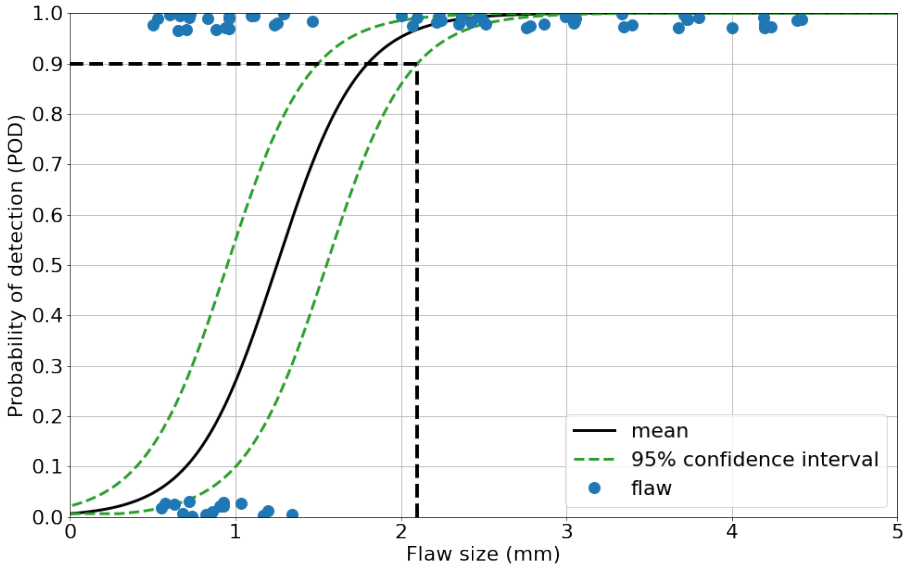


Figure 4.2: Illustration of the probability of detection (POD) curve. The intersection of the black dashed lines determines the flaw size that has a probability of detection of 90 % with the 95% confidence ($a_{90/95}$)

in this thesis. Instead, a mean average precision was used, but with some additional analysis done in Pub 1 that showed the reliability of the proposed methods and enabled the readers to clearly see how many of the defect's appearances were detected.

Chapter 5

Main scientific contributions of the thesis

The main scientific contributions of this thesis are: (I) deep learning-based method for detection of defects with extreme aspect ratio with results disseminated in [Pub1], [Pub2], and [Pub3]; (II) deep learning-based method for defect detection by simultaneous analysis of multiple ultrasound images with results disseminated in [Pub4], and [Pub5].

5.1 Deep learning-based method for detection of defects with extreme aspect ratios

Deep learning object detectors achieve good results when applied to general object detection on natural scenes. The techniques like transfer learning [118] and data augmentation [119], allowed researchers to leverage the good performance of these detectors and apply them to other domains. However, the application of deep learning object detectors for defect detection in ultrasonic images was hindered by the lack of realistic datasets. A thorough search of the relevant literature did not yield any research articles before [30] that applied deep learning object detectors for defect detection from ultrasonic images. Usage of deep learning is fairly new in this domain, and the lack of public datasets prevents realistic comparison of the published methods. As stated earlier, the focus of this thesis is on the application and development of one-stage object detectors for defect detection from ultrasonic images. The first step in this process was to establish baseline results and compare several of the top-performing object detections on a large database of ultrasonic B-scans. This was one of the contributions of [Pub1]. A thorough evaluation enabled comparison among the existing methods and gave insight into the reliability of object detection methods when applied in UT data analysis. An additional contribution of this work was a procedure for calculating the anchors' hyperparameters. In Section 3.2 it was shown that anchors' design has a huge impact on object detector training, and the proper setup of anchors can have a positive effect on the detector's performance. The importance of choosing the right anchors is highlighted when the objects that need to be detected have extreme

aspect ratios, which is common in phased array ultrasonic (PAUT) images. If the default anchor settings are used, the training can be difficult and the models sometimes do not even converge. However, if the procedure from [Pub1] is used to calculate anchors' hyperparameters, an improvement of almost 6% is achieved compared to the default model. Other valuable insights, that revealed new research directions, were also presented in this publication. It was shown that the smaller networks outperform bigger ones, which indicated that further improvement might be achievable if more data was available or even simpler architecture was used. These two hypotheses were explored in the follow-up works ([Pub3], and [Pub2]).

Since the acquisition of additional ultrasonic data is very expensive, in [Pub3] the additional data was synthetically generated. While some previous works artificially generated UT data, the used approaches were fairly simple and relied on copy-pasting of the defects into empty background images. A better quality of the generated images can be achieved if a generative adversarial network (GAN) is used. Additionally, if those images are to be used for improving the precision of a deep learning object detector, it is useful to make certain modifications of the standard GAN architecture. In [Pub3] it was shown that the GAN with additional object detection discriminator network can be used to generate realistic new B-scans. Moreover, the generated B-scans can be used as additional data when training an object detector and improve the mean average precision by almost 6 %.

The second mentioned insight, regarding the possible benefits achieved by the simplification of the used neural network, was explored in [Pub2]. A novel encoder-decoder-based feature extractor was designed and implemented while keeping in mind insights provided in [Pub1]. A small number of parameters enables easier training on a small dataset and reduces computational complexity. The skip connections used in the network minimize the loss of information that was noticed in [Pub1] by comparing the results of object detectors on various input images resolution. Furthermore, the feature network and the detection head of the architecture proposed in [Pub2], were designed to enable dense placement of anchors on the x-axis of feature maps. This modification was proposed because some anchors calculated with the procedure from [Pub1] had aspect ratio so extreme that the used anchors did not overlap and properly cover the entire image. A DefectDet architecture proposed in [Pub2] achieved additional improvements of both precision and inference time compared to the baseline EfficientDet-D0 model. It was also shown that the proposed network outperforms other state-of-the-art architecture such as YOLOv5 in terms of mean average precision.

5.2 Deep learning-based method for defect detection by simultaneous analysis of multiple ultrasound images

Due to the nature of the acquired PAUT images, useful information for detecting defects is often found across multiple B-scans. It is expected from the inspectors viewing monotone sequences of images to often rewind and get a better view of suspicious signals. The inspectors confirm their decision by looking at the same area of the material from various angles and scanning directions. To ensure the reliability of the UT inspection, the data must be acquired from various scanning angles and all of the data must be inspected. This prevents the cases where the transmitted ultrasonic waves propagate parallel to a flat defect, which would result in no reflection of the waves. This would ultimately mean the signal for this defect would not be seen, and it would be missed during the analysis. The data is usually acquired for dozens of angles. The dataset in this thesis, for example, has angles ranging from 45° up to 79° with a two-degree increment. This also means a huge amount of similar data must be analyzed, regardless of the way the analysis is performed (manual or automated).

In [Pub4] a novel approach for simultaneous analysis of UT B-scans acquired from different scanning angles is proposed. The method from this publication is proposed to reduce the time needed for the overall automated analysis by merging the data acquired for different scanning angles. The method relies on the attention mechanism that determines which of the input angles are the most useful, and then the images from those angles are given higher importance during the data fusion. Since the images from different angles are fused inside the model, the automated inspection can be performed in a number of steps equal to the number of the unique cross-section. This means a significant ($\sim$15 times) reduction of time needed for the analysis in a real-life setting.

A different approach for analyzing sequences of ultrasonic images was presented in [Pub5]. The goal of the methods proposed in this publication was to improve the precision of a defect detection model by using the additional information. The methods from [Pub1],[Pub2],[Pub3] all rely on independent B-scan analysis. However, this strategy focuses only on the defect's visual similarity and ignores the "temporal" consistency. The temporal consistency, in this case, refers to the dynamical alteration of the defect's signal across neighboring cross-sections of the material. In [Pub5], it was first shown that the simple expansion of the input does not work well and that a more advanced approach is needed. Two novel methods were then proposed to enable a one-stage object detector to leverage the additional context available when the sequence of consecutive B-scan is analyzed. This was implemented by passing the consecutive B-scans through a feature extractor and feature pyramid network, and then combining the obtained feature maps. For each input image, five feature maps of different resolutions are calculated. Calculated feature maps contain high-dimensional information about the content

in each of inputted B-scans. These feature maps are then combined using either a standard convolutional layer or a combination of the convolutional layer with a long short-term memory layer (ConvLSTM). Both of the proposed approaches work well and lead to a significant mean average precision improvement compared to the standard EfficientDet model.

Chapter 6

List of publications

Pub 1 **D. Medak**, L. Posilović, M. Subašić, M. Budimir, S. Lončarić, ”Automated Defect Detection From Ultrasonic Images Using Deep Learning”, *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 68, no. 10, Oct. 2021, pp. 3126-3134, doi: 10.1109/TUFFC.2021.3081750.

Pub 2 **D. Medak**, L. Posilović, M. Subašić, M. Budimir, S. Lončarić, ”DefectDet: a deep learning architecture for detection of defects with extreme aspect ratios in ultrasonic images”, *Neurocomputing*, vol. 473, Feb. 2022, pp. 107-115, doi: 10.1016/j.neucom.2021.12.008.

Pub 3 L. Posilović, **D. Medak**, M. Subašić, M. Budimir, S. Lončarić, ”Generative adversarial network with object detector discriminator for enhanced defect detection on ultrasonic b-scans”, *Neurocomputing*, vol. 459, Oct. 2021, pp. 361-369, doi: 10.1016/j.neucom.2021.06.094.

Pub 4 **D. Medak**, L. Posilović, M. Subašić, T. Petković, M. Budimir, S. Lončarić, ”Rapid Defect Detection by Merging Ultrasound B-scans from Different Scanning Angles”, in *Proc. of the 12th International Symposium on Image and Signal Processing and Analysis (ISPA)*, Sep. 2021, pp. 219-224, doi: 10.1109/ISPA52656.2021.9552050.

Pub 5 **D. Medak**, L. Posilović, M. Subašić, M. Budimir, S. Lončarić, ”Deep learning-based defect detection from sequences of ultrasonic B-scans”, *IEEE Sensors*, Dec. 2021, doi: 10.1109/JSEN.2021.3134452.

Chapter 7

Author's contribution to the publications

The results presented in this thesis are based on the research carried out during the period of 2018-2022 at the University of Zagreb Faculty of Electrical Engineering and Computing, mostly as a part of the research project SMART UTX: Smart modular system for ultrasound diagnostics in extreme conditions. This research was co-funded by the European Union through the European Regional Development Fund, under the grant KK.01.2.1.01.0151 (Smart UTX). The thesis includes five publications written in collaboration with the coauthors of the published papers. The author's contribution to each paper consists of the conceptualization of novel methods, data curation and preparation, software implementation, performing the required experiments, results analysis, text writing, and presentation.

[Pub1] In the paper **"Automated Defect Detection From Ultrasonic Images Using Deep Learning"** the author proposed usage of deep learning object detectors for analysis of UT images following conclusions from several related works about the superiority of deep learning approaches compared to traditional approaches. An important part of the work was to collect and annotate the largest database of real ultrasonic B-scans that was until then used in the literature. This was done to ensure the credibility of reported results. Upon manual annotation of over 4000 images, the author implemented several state-of-the-art object detectors and compared their performances. A novel method inspired by previous work was proposed to calculate networks' hyperparameters related to anchors' shapes. Using this method, a significant improvement of mean average precision (MAP) was achieved. Finally, to prove the reliability of the proposed method, a thorough evaluation was performed.

[Pub2] In the paper **"DefectDet: a deep learning architecture for detection of defects with extreme aspect ratios in ultrasonic images"** the author proposed a novel deep learning architecture for defect detection in ultrasonic B-scans. This work was done as a continuation of [Pub1], and solutions to several previously noticed shortcomings were proposed. First, to tackle the problem of a small dataset, a novel simpler feature extractor based on an encoder-decoder network was proposed and implemented by the author. A novel feature extractor improved

mean average precision while simultaneously reducing inference time. Additionally, the new backbone reduced the loss of information when analyzing images of smaller resolution. The author also proposed a novel detection head that was designed to improve the performance of the detector when detecting objects with extreme aspect ratios, such as defects in ultrasonic B-scans. It was shown that both of these components independently lead to improved MAP, but the merits were even bigger once the two components were merged into a new deep learning network that was named DefectDet.

[Pub3] In the paper **"Generative adversarial network with object detector discriminator for enhanced defect detection on ultrasonic b-scans"** a different approach was used to tackle the problem of small datasets of UT B-scans. A generative adversarial network (GAN) was used to artificially expand the dataset by generating B-scans that contained defects on positions defined by the input masks. Generated images were then used in combination with the original image to improve the MAP of the YOLOv3 object detector. In order to develop GAN that generates images that are useful for object detection training, YOLOv3 was used as an additional discriminator during the GAN training. The author's contribution to this work includes the development of a new object detector, performing experiments, and paper reviewing and editing.

[Pub4] In the paper **"Rapid Defect Detection by Merging Ultrasound B-scans from Different Scanning Angles"** a novel approach for simultaneous analysis of B-scans acquired at different angles is proposed by the author. The main motivation behind analyzing images from multiple scanning angles is to improve the precision or to reduce the overall needed time for data analysis. Even when the UT data analysis is performed in an automated fashion using some algorithm that is run on the computer, it can still take a long time if the amount of data is huge. This is often the case with phased array data, where the scans are acquired from many different angles. The author proposed a new model that uses an attention mechanism to determine which of the input angles the object detectors should focus on. The input images are then merged in a way that preserves information from scans for which the model previously determined that are more important. It was shown that the proposed approach analyses UT data achieves similar precision and it is around 15 times faster compared to the traditional approach where the images are analyzed independently.

[Pub5] In the paper **"Deep learning-based defect detection from sequences of ultrasonic B-scans"** the author proposed two novel methods for analyzing sequences of UT B-scans. The proposed architectures were designed to enable object detectors to look at the surrounding area of some B-scan. Human inspectors also do this when performing the analysis since it enables them to confirm their decision. The defect usually spans across several B-scans that display neighboring cross-sections of the inspected material. However, a simple expansion of the deep neural network input to work with several neighboring B-scans does not lead to improvement.

More complex approaches that are based on high-dimensional feature maps merging are needed and their usage improves MAP. The author designed and implemented two of such methods and experimentally proved the benefits of their application.

Chapter 8

Conclusions and future directions

Currently used procedures for ultrasonic testing data analysis still rely mostly on the knowledge and experience of the human inspectors. It takes years of practice and training for a human operator to acquire the skills needed to perform the analysis of the UT data. Even then, the decision made by humans can be subjective and prone to error, especially in cases where a large amount of data needs to be analyzed, which leads to fatigue of the inspectors. The procedure used by the inspectors can not be explicitly expressed as a set of rules, which makes the development of methods for automated UT data analysis difficult. The development of methods for automated analysis flourished recently due to many improvements made in the deep learning area. Deep learning methods are very promising in this field since they can implicitly learn to detect defects by training on large amounts of labeled images. Their generalization abilities are a lot better than those of the traditional approaches and in some works, their performance was on par with the human-level performance. However, directly applying the existing deep learning architectures for this task will not enable the usage of the full potential of deep learning models. In a series of publications attached to this thesis, novel deep learning object detectors were designed, taking into account the application domain. A thorough evaluation was performed to prove the merits of each individual solution. The novel models, components, design choices, and training procedures that are proposed in these publications can also be used jointly. This enables the creation of the ultimate UT defect detector, which is lightweight, fast, reliable, works well with the objects of extreme aspect ratios, and is able to use additional context when detecting defects.

While the contributions presented in this thesis bring the automated analysis of UT data to a new level, there is still room for progress. The advances of deep learning methods in the NDE domain will probably come by improving the three building blocks of deep learning-based defect detection: the data, the used method, and the evaluation procedure. In this work, calibration blocks were scanned to acquire the UT data that was used for the development and evaluation of automated analysis methods. There are many other common use cases where deep learning could be applied such as the analysis of bolts, welds, pipelines, etc. Also, the

information inputted into an automated analysis model can be enriched by providing the actual positions in 3D space, or by using the data acquired from different directions (skews). Another option is to fuse the UT data with the data obtained by some other NDE technique. The second direction of improvement should be focused on the new method. This can be either in the form of introducing new tasks such as anomaly detection, next frame prediction, 3D data generation, and similar, or by applying a novel method for some existing task like the usage of transformers networks for detection of defects. Further research in this area will enable the application of state-of-the-art deep learning models and techniques and can lead to further improvement. Finally, to objectively measure the achieved improvement, a suitable metric must be considered. Currently used metrics such as ROC, MAP, and POD all have some disadvantages when used independently to evaluate an automated method for image analysis in the NDE domain. This was also noticed by other researchers and with the development of NDE 4.0 proper evaluation of AI-based solutions will require more attention. In this thesis, the MAP was used for evaluation, but additional metrics such as POD should be calculated and the proposed methods should be tested in a real environment before they can be used to assist the human experts in the field inspections.

Bibliography

[1]Hellier, C., Handbook of Nondestructive Evaluation, 3E. McGraw-Hill Education, 2020, available at: https://books.google.hr/books?id=rzDjygEACAAJ

[2]Cartz, L., Nondestructive testing: radiography, ultrasonics, liquid penetrant, magnetic particle, eddy current. ASM International, 1995, available at: https://books.google.hr/books?id=0spRAAAAMAAJ

[3]Sotero, R., Albuquerque, M., Paula, F., Farias, C., Simas Filho, E., "Classification of ultrasonic signs pre-processed by fourier transform through artificial neural network using the echo pulse technique for the identification of defects in welded joints of structural steel", Journal of Mechanics Engineering and Automation, Vol. 5, 05 2015.

[4]Latête, T., Gauthier, B., Belanger, P., "Towards using convolutional neural network to locate, identify and size defects in phased array ultrasonic testing", Ultrasonics, Vol. 115, 2021, p. 106436, available at: https://www.sciencedirect.com/science/article/pii/S0041624X21000731

[5]Vishal, V., Ramya, R., Vinay Srinivas, P., Vimal Samsingh, R., "A review of implementation of artificial intelligence systems for weld defect classification", Materials Today: Proceedings, Vol. 16, 2019, pp. 579–583, international Conference on Advances in Materials, Manufacturing and Applied Sciences, available at: https://www.sciencedirect.com/science/article/pii/S2214785319309769

[6]Bettayeb, F., Rachedi, T., Benbartaoui, H., "An improved automated ultrasonic nde system by wavelet and neuron networks", Ultrasonics, Vol. 42, No. 1, 2004, pp. 853–858, proceedings of Ultrasonics International 2003, available at: https://doi.org/10.1016/j.ultras.2004.01.064

[7]Matz, V., Kreidl, M., Smid, R., "Classification of ultrasonic signals", International Journal of Materials, Vol. 27, 10 2006, pp. 145-, available at: https://doi.org/10.1504/IJMPT.2006.011267

[8]Sambath, S., Nagaraj, P., Selvakumar, N., “Automatic defect classification in ultrasonic ndt using artificial intelligence”, Journal of Nondestructive Evaluation, Vol. 30, No. 1, Mar 2011, pp. 20–28, available at: https://doi.org/10.1007/s10921-010-0086-0

[9]Al-Ataby, A., Al-Nuaimy, W., Brett, C., Zahran, O., “Automatic detection and classification of weld flaws in tofd data using wavelet transform and support vector machines”, Insight - Non-Destructive Testing and Condition Monitoring, Vol. 52, 11 2010, pp. 597–602, available at: https://doi.org/10.1784/insi.2010.52.11.597

[10]Chen, Y., Ma, H.-W., Zhang, G.-M., “A support vector machine approach for classification of welding defects from ultrasonic signals”, Nondestructive Testing and Evaluation, Vol. 29, No. 3, 2014, pp. 243–254, available at: https://doi.org/10.1080/10589759.2014.914210

[11]Meng, M., Chua, Y. J., Wouterson, E., Ong, C. P. K., “Ultrasonic signal classification and imaging system for composite materials via deep convolutional neural networks”, Neurocomputing, Vol. 257, 2017, pp. 128–135, machine Learning and Signal Processing for Big Multimedia Analysis, available at: https://doi.org/10.1016/j.neucom.2016.11.066

[12]Cruz, F., Filho, E. S., Albuquerque, M., Silva, I., Farias, C., Gouvêa, L., “Efficient feature selection for neural network based detection of flaws in steel welded joints using ultrasound testing”, Ultrasonics, Vol. 73, 2017, pp. 1–8, available at: https://doi.org/10.1016/j.ultras.2016.08.017

[13]Khelil, M., Boudraa, M., Kechida, A., Drai, R., “Classification of Defects by the SVM Method and the Principal Component Analysis (PCA)”, available at: https://doi.org/10.5281/zenodo.1060751 09 2007.

[14]Veiga, J., A. de Carvalho, A., Silva, I., M. A. Rebello, J., “The use of artificial neural network in the classification of pulse-echo and tofd ultrasonic signals”, Journal of The Brazilian Society of Mechanical Sciences and Engineering - J BRAZ SOC MECH SCI ENG, Vol. 27, 10 2005, available at: https://doi.org/10.1590/S1678-58782005000400007

[15]Munir, N., Kim, H.-J., Song, S.-J., Kang, S.-S., “Investigation of deep neural network with drop out for ultrasonic flaw classification in weldments”, Journal of Mechanical Science and Technology, Vol. 32, No. 7, Jul 2018, pp. 3073–3080, available at: https://doi.org/10.1007/s12206-018-0610-1

[16]Munir, N., Kim, H.-J., Park, J., Song, S.-J., Kang, S.-S., “Convolutional neural network for ultrasonic weldment flaw classification in noisy conditions”, Ultrasonics,

Vol. 94, 2019, pp. 74–81, available at: http://www.sciencedirect.com/science/article/pii/S0041624X18305754

[17]Munir, N., Park, J., Kim, H.-J., Song, S.-J., Kang, S.-S., "Performance enhancement of convolutional neural network for ultrasonic flaw classification by adopting autoencoder", NDT & E International, Vol. 111, 2020, p. 102218, available at: https://www.sciencedirect.com/science/article/pii/S0963869519306243

[18]Kechida, A., Drai, R., Guessoum, A., "Texture analysis for flaw detection in ultrasonic images", Journal of Nondestructive Evaluation, Vol. 31, No. 2, Jun 2012, pp. 108–116, available at: https://doi.org/10.1007/s10921-011-0126-4

[19]Cygan, H., Girardi, L., Aknin, P., Simard, P., "B-scan ultrasonic image analysis for internal rail defect detection", in World Congress on Railway Research, 10 2003.

[20]Petcher, P., Dixon, S., "Parabola detection using matched filtering for ultrasound b-scans.", Ultrasonics, Vol. 52 1, 2012, pp. 138-44.

[21]Kieckhoefer, H., Baan, J., Mast, A., Volker, W. A., "Image processing techniques for ultrasonic inspection", in Proc. 17th World Conference on Nondestructive Testing, Shanghai, China, 2008.

[22]Pyle, R. J., Bevan, R. L. T., Hughes, R. R., Rachev, R. K., Ali, A. A. S., Wilcox, P. D., "Deep learning for ultrasonic crack characterization in nde", IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control, 2020, pp. 1–1.

[23]Ren, S., He, K., Girshick, R., Sun, J., "Faster r-cnn: Towards real-time object detection with region proposal networks", in Advances in Neural Information Processing Systems 28, Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., Garnett, R., (ur.). Curran Associates, Inc., 2015, pp. 91–99, available at: http://papers.nips.cc/paper/5638-faster-r-cnn-towards-real-time-object-detection-with-region-proposal-networks.pdf

[24]Virkkunen, I., Koskinen, T., Jessen-Juhler, O., Rinta-aho, J., "Augmented ultrasonic data for machine learning", Journal of Nondestructive Evaluation, Vol. 40, No. 1, 2021, pp. 1–11.

[25]Simonyan, K., Zisserman, A., "Very deep convolutional networks for large-scale image recognition", CoRR, Vol. abs/1409.1556, 2014, available at: https://arxiv.org/abs/1409.1556

[26]Annis, C., "Nondestructive Evaluation System Reliability Assessment", available at: http://statisticalengineering.com/mh1823/ 04 2009.

[27]American Society for Testing and Materials, “Astm: Standard practice for probability of detection analysis for hit/miss data, astm e2862-12”, 2012.

[28]Siljama, O., Koskinen, T., Jessen-juhler, O., Virkkunen, I., “Automated flaw detection in multi-channel phased array ultrasonic data using machine learning”, Journal of Nondestructive Evaluation, Vol. 40, No. 3, Aug. 2021, funding Information: Welds were contributed by Suisto Engineering. UT data scanning was contributed by DEKRA. Data augmentation was contributed by Trueflaw. Their support is gratefully acknowledged. Publisher Copyright: © 2021, The Author(s).

[29]Virupakshappa, K., Oruklu, E., “Multi-class classification of defect types in ultrasonic ndt signals with convolutional neural networks”, in 2019 IEEE International Ultrasonics Symposium (IUS), 2019, pp. 1647–1650.

[30]Posilovi ć, L., Medak, D., Subašić, M., Petković, T., Budimir, M., Lončarić, S., “Flaw detection from ultrasonic images using yolo and ssd”, in 2019 11th International Symposium on Image and Signal Processing and Analysis (ISPA). IEEE, 2019, pp. 163–168.

[31]Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S. E., Fu, C., Berg, A. C., “SSD: single shot multibox detector”, Vol. abs/1512.02325, 10 2015, pp. 21–37, available at: https://doi.org/10.1007/978-3-319-46448-0%5F2

[32]Redmon, J., Farhadi, A., “Yolov3: An incremental improvement”, 2018, cite arxiv:1804.02767Comment: Tech Report, available at: https://arxiv.org/abs/1804.02767

[33]Provencal, E., Laperrière, L., “Identification of weld geometry from ultrasound scan data using deep learning”, Procedia CIRP, Vol. 104, 2021, pp. 122-127, 54th CIRP CMS 2021 - Towards Digitalized Manufacturing 4.0, available at: https://www.sciencedirect.com/science/article/pii/S2212827121009197

[34]Park, S.-H., Hong, J.-Y., Ha, T., Choi, S., Jhang, K.-Y., “Deep learning-based ultrasonic testing to evaluate the porosity of additively manufactured parts with rough surfaces”, Metals, Vol. 11, No. 2, 2021, available at: https://www.mdpi.com/2075-4701/11/2/290

[35]Cha, Y.-J., Choi, W., Büyüköztürk, O., “Deep learning-based crack damage detection using convolutional neural networks”, Computer-Aided Civil and Infrastructure Engineering, Vol. 32, No. 5, 2017, pp. 361–378, available at: https://onlinelibrary.wiley.com/doi/abs/10.1111/mice.12263

[36]Faghih-Roohi, S., Hajizadeh, S., Núñez, A., Babuska, R., De Schutter, B., “Deep convolutional neural networks for detection of rail surface defects”, in 2016 International Joint Conference on Neural Networks (IJCNN), 2016, pp. 2584–2589.

[37]Yi, X., Song, Y., Zhang, Y., “Enhanced darknet53 combine mlfpn based real-time defect detection in steel surface”, in Chinese Conference on Pattern Recognition and Computer Vision (PRCV). Springer, 2020, pp. 303–314.

[38]Fu, G., Sun, P., Zhu, W., Yang, J., Cao, Y., Yang, M. Y., Cao, Y., “A deep-learning-based approach for fast and robust steel surface defects classification”, Optics and Lasers in Engineering, Vol. 121, 2019, pp. 397–405, available at: https://www.sciencedirect.com/science/article/pii/S0143816619301678

[39]Bastian, B. T., N, J., Ranjith, S. K., Jiji, C., “Visual inspection and characterization of external corrosion in pipelines using deep neural network”, NDT & E International, Vol. 107, 2019, p. 102134, available at: https://www.sciencedirect.com/science/article/pii/S096386951930060X

[40]Yu, Y., Cao, H., Yan, X., Wang, T., Ge, S. S., “Defect identification of wind turbine blades based on defect semantic features with transfer feature extractor”, Neurocomputing, Vol. 376, 2020, pp. 1–9, available at: http://www.sciencedirect.com/science/article/pii/S0925231219313396

[41]Dalal, N., Triggs, B., “Histograms of oriented gradients for human detection”, in international Conference on computer vision & Pattern Recognition (CVPR’05), Vol. 1. IEEE Computer Society, 2005, pp. 886–893.

[42]Lowe, D. G., “Distinctive image features from scale-invariant keypoints”, Int. J. Comput. Vision, Vol. 60, No. 2, Nov. 2004, pp. 91–110, available at: https://doi.org/10.1023/B:VISI.0000029664.99615.94

[43]Masci, J., Meier, U., Ciresan, D., Schmidhuber, J., Fricout, G., “Steel defect classification with max-pooling convolutional neural networks”, in The 2012 International Joint Conference on Neural Networks (IJCNN), 2012, pp. 1–6.

[44]Woo, S., Park, J., Lee, J.-Y., Kweon, I. S., “Cbam: Convolutional block attention module”, in Computer Vision – ECCV 2018, Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y., (ur.). Cham: Springer International Publishing, 2018, pp. 3–19, available at: https://doi.org/10.1007/978-3-030-01234-2_1

[45]Lin, T.-Y., DollÃ!’r, P., Girshick, R. B., He, K., Hariharan, B., Belongie, S. J., “Feature pyramid networks for object detection”, CoRR, Vol. abs/1612.03144, 2016, available at: http://arxiv.org/abs/1612.03144

[46]Song, X., Chen, K., Cao, Z., “Resnet-based image classification of railway shelling defect”, in 2020 39th Chinese Control Conference (CCC), 2020, pp. 6589-6593.

[47]Luo, Q., Gao, B., Woo, W., Yang, Y., “Temporal and spatial deep learning network for infrared thermal defect detection”, NDT & E International, Vol. 108, 2019, p. 102164, available at: http://www.sciencedirect.com/science/article/pii/S0963869519301355

[48]Ronneberger, O., Fischer, P., Brox, T., “U-net: Convolutional networks for biomedical image segmentation”, in Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, Navab, N., Hornegger, J., Wells, W. M., Frangi, A. F., (ur.), 2015, pp. 234–241.

[49]Hochreiter, S., Schmidhuber, J., “Long short-term memory”, Neural computation, Vol. 9, No. 8, 1997, pp. 1735–1780.

[50]Ruan, L., Gao, B., Wu, S., Woo, W. L., “Deftectnet: Joint loss structured deep adversarial network for thermography defect detecting system”, Neurocomputing, Vol. 417, 2020, pp. 441–457, available at: http://www.sciencedirect.com/science/article/pii/S0925231220312637

[51]Bang, H.-T., Park, S., Jeon, H., “Defect identification in composite materials via thermography and deep learning techniques”, Composite Structures, Vol. 246, 2020, p. 112405, available at: http://www.sciencedirect.com/science/article/pii/S026382232030146X

[52]Du, W., Shen, H., Fu, J., Zhang, G., He, Q., “Approaches for improvement of the x-ray image defect detection of automobile casting aluminum parts based on deep learning”, NDT & E International, Vol. 107, 2019, p. 102144, available at: http://www.sciencedirect.com/science/article/pii/S0963869519300192

[53]Le, X., Mei, J., Zhang, H., Zhou, B., Xi, J., “A learning-based approach for surface defect detection using small image datasets”, Neurocomputing, Vol. 408, 2020, pp. 112–120, available at: http://www.sciencedirect.com/science/article/pii/S0925231220303386

[54]Arjovsky, M., Chintala, S., Bottou, L., “Wasserstein generative adversarial networks”, in Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017, ser. Proceedings of Machine Learning Research, Precup, D., Teh, Y. W., (ur.), Vol. 70. PMLR, 2017, pp. 214–223, available at: http://proceedings.mlr.press/v70/

[55]Szegedy, C., Ioffe, S., Vanhoucke, V., “Inception-v4, inception-resnet and the impact of residual connections on learning”, CoRR, Vol. abs/1602.07261, 2016, available at: http://arxiv.org/abs/1602.07261

[56]Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H., "Mobilenets: Efficient convolutional neural networks for mobile vision applications", CoRR, Vol. abs/1704.04861, 2017, available at: http://arxiv.org/abs/1704.04861

[57]Wang, Y., Guo, H., "Weld defect detection of x-ray images based on support vector machine", IETE Technical Review, Vol. 31, No. 2, 2014, pp. 137-142, available at: https://doi.org/10.1080/02564602.2014.892739

[58]Ren, J., Ren, R., Green, M., Huang, X., "Defect detection from x-ray images using a three-stage deep learning algorithm", in 2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE), 2019, pp. 1-4.

[59]Lin, T., Maire, M., Belongie, S. J., Bourdev, L. D., Girshick, R. B., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C. L., "Microsoft COCO: common objects in context", in Computer Vision – ECCV 2014, Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T., (ur.), Vol. abs/1405.0312. Cham: Springer International Publishing, 2014, pp. 740–755, available at: https://doi.org/10.1007/978-3-319-10602-1%5F48

[60]Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., Fei-Fei, L., "ImageNet Large Scale Visual Recognition Challenge", International Journal of Computer Vision (IJCV), Vol. 115, No. 3, 2015, pp. 211–252.

[61]Everingham, M., van Gool, L., Williams, C., Winn, J., Zisserman, A., "The PASCAL Object Recognition Database Collection.", http://host.robots.ox.ac.uk/pascal/VOC/, [Online; accessed 1-May-2020]. 2012.

[62]Krizhevsky, A., Sutskever, I., Hinton, G. E., "Imagenet classification with deep convolutional neural networks", in NIPS, 2012.

[63]Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., Xie, S., "A convnet for the 2020s", arXiv preprint arXiv:2201.03545, 2022.

[64]Lecun, Y., Bottou, L., Bengio, Y., Haffner, P., "Gradient-based learning applied to document recognition", Proceedings of the IEEE, Vol. 86, No. 11, 1998, pp. 2278–2324.

[65]Hinton, G., "Neural networks for machine learning lecture 6", Online Lecture, available at: http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf 2018.

[66]Kingma, D. P., Ba, J., "Adam: A method for stochastic optimization", arXiv preprint arXiv:1412.6980, 2014.

[67]Zeiler, M. D., “Adadelta: an adaptive learning rate method”, arXiv preprint arXiv:1212.5701, 2012.

[68]Duchi, J., Hazan, E., Singer, Y., “Adaptive subgradient methods for online learning and stochastic optimization”, Journal of Machine Learning Research, Vol. 12, No. 61, 2011, pp. 2121-2159, available at: http://jmlr.org/papers/v12/duchi11a.html

[69]Ioffe, S., Szegedy, C., “Batch normalization: Accelerating deep network training by reducing internal covariate shift”, in International conference on machine learning. PMLR, 2015, pp. 448–456.

[70]Law, H., Deng, J., “Cornernet: Detecting objects as paired keypoints”, in Proceedings of the European conference on computer vision (ECCV), 2018, pp. 734–750.

[71]Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q., Tian, Q., “Centernet: Keypoint triplets for object detection”, in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 6569–6578.

[72]Tian, Z., Shen, C., Chen, H., He, T., “Fcos: Fully convolutional one-stage object detection”, in Proceedings of the IEEE/CVF international conference on computer vision, 2019, pp. 9627–9636.

[73]Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., Polosukhin, I., “Attention is all you need”, in Advances in neural information processing systems, 2017, pp. 5998–6008.

[74]Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S., “End-to-end object detection with transformers”, in European Conference on Computer Vision. Springer, 2020, pp. 213–229.

[75]Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B., “Swin transformer: Hierarchical vision transformer using shifted windows”, arXiv preprint arXiv:2103.14030, 2021.

[76]Tan, M., Pang, R., Le, Q. V., “Efficientdet: Scalable and efficient object detection”, ArXiv, Vol. abs/1911.09070, 2019.

[77]Bochkovskiy, A., Wang, C.-Y., Liao, H.-Y. M., “Yolov4: Optimal speed and accuracy of object detection”, ArXiv, Vol. abs/2004.10934, 2020.

[78]He, K., 0006, X. Z., Ren, S., 0001, J. S., “Deep residual learning for image recognition”, CoRR, Vol. abs/1512.03385, 2015, available at: http://arxiv.org/abs/1512.03385

[79]Xie, S., Girshick, R. B., Dollár, P., Tu, Z., He, K., “Aggregated residual transformations for deep neural networks”, CoRR, Vol. abs/1611.05431, 2016, available at: http://arxiv.org/abs/1611.05431

[80]Sandler, M., Howard, A. G., Zhu, M., Zhmoginov, A., Chen, L., “Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation”, CoRR, Vol. abs/1801.04381, 2018, available at: http://arxiv.org/abs/1801.04381

[81]Howard, A., Sandler, M., Chu, G., Chen, L.-C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., Le, Q. V., Adam, H., “Searching for mobilenetv3”, in The IEEE International Conference on Computer Vision (ICCV), October 2019.

[82]Huang, G., Liu, Z., Weinberger, K. Q., “Densely connected convolutional networks”, CoRR, Vol. abs/1608.06993, 2016, available at: http://arxiv.org/abs/1608.06993

[83]Tan, M., Le, Q. V., “Efficientnet: Rethinking model scaling for convolutional neural networks”, CoRR, Vol. abs/1905.11946, 2019, available at: http://arxiv.org/abs/1905.11946

[84]0005, S. L., Qi, L., Qin, H., Shi, J., Jia, J., “Path aggregation network for instance segmentation”, CoRR, Vol. abs/1803.01534, 2018, available at: http://arxiv.org/abs/1803.01534

[85]Girshick, R., “Fast r-cnn”, in Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), ser. ICCV ’15. USA: IEEE Computer Society, 2015, p. 1440–1448, available at: https://doi.org/10.1109/ICCV.2015.169

[86]Ren, S., He, K., Girshick, R. B., 0001, J. S., “Faster r-cnn: Towards real-time object detection with region proposal networks”, IEEE Trans. Pattern Anal. Mach. Intell, Vol. 39, No. 6, 2017, pp. 1137–1149, available at: http://doi.ieeecomputersociety.org/10.1109/TPAMI.2016.2577031

[87]Dai, J., Li, Y., He, K., 0001, J. S., “R-fcn: Object detection via region-based fully convolutional networks”, CoRR, Vol. abs/1605.06409, 2016, available at: http://arxiv.org/abs/1605.06409

[88]He, K., Gkioxari, G., Dollár, P., Girshick, R. B., “Mask r-cnn”, 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2980–2988.

[89]He, K., Zhang, X., Ren, S., Sun, J., “Spatial pyramid pooling in deep convolutional networks for visual recognition”, IEEE transactions on pattern analysis and machine intelligence, Vol. 37, No. 9, 2015, pp. 1904–1916.

[90]Redmon, J., Divvala, S., Girshick, R., Farhadi, A., "You only look once: Unified, real-time object detection", in CVPR. IEEE Computer Society, 2016, pp. 779–788, available at: https://arxiv.org/abs/1506.02640

[91]Fu, C., Liu, W., Ranga, A., Tyagi, A., Berg, A. C., "DSSD : Deconvolutional single shot detector", CoRR, Vol. abs/1701.06659, 2017, available at: http://arxiv.org/abs/1701.06659

[92]Lin, T., Goyal, P., Girshick, R. B., He, K., Dollár, P., "Focal loss for dense object detection", CoRR, Vol. abs/1708.02002, 2017, available at: http://arxiv.org/abs/1708.02002

[93]Zhang, S., Wen, L., Bian, X., Lei, Z., Li, S. Z., "Single-shot refinement neural network for object detection", in CVPR, 2018.

[94]Redmon, J., Farhadi, A., "Yolo9000: Better, faster, stronger", CoRR, Vol. abs/1612.08242, 2016, available at: http://arxiv.org/abs/1612.08242

[95]Jocher, G., "ultralytics/yolov5: v3.1 - Bug Fixes and Performance Improvements", available at: https://doi.org/10.5281/zenodo.4154370 Oct. 2020.

[96]Zlocha, M., Dou, Q., Glocker, B., "Improving retinanet for ct lesion detection with dense masks from weak recist labels", arXiv preprint arXiv:1906.02283, 2019.

[97]Ahmad, M., Abdullah, M., Han, D., "Small object detection in aerial imagery using retinanet with anchor optimization", in 2020 International Conference on Electronics, Information, and Communication (ICEIC), 2020, pp. 1–3.

[98]Zhong, Y., Wang, J., Peng, J., Zhang, L., "Anchor box optimization for object detection", 2020 IEEE Winter Conference on Applications of Computer Vision (WACV), 2020, pp. 1275–1283.

[99]Liu, L., Muelly, M., Deng, J., Pfister, T., Li, L.-J., "Generative modeling for small-data object detection", in 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 6072-6080.

[100]Han, C., Kitamura, Y., Kudo, A., Ichinose, A., Rundo, L., Furukawa, Y., Umemoto, K., Li, Y., Nakayama, H., "Synthesizing diverse lung nodules wherever massively: 3d multi-conditional gan-based ct image augmentation for object detection", in 2019 International Conference on 3D Vision (3DV), 2019, pp. 729-737.

[101]Han, W., Khorrami, P., Paine, T. L., Ramachandran, P., Babaeizadeh, M., Shi, H., Li, J., Yan, S., Huang, T. S., "Seq-nms for video object detection", arXiv preprint arXiv:1602.08465, 2016.

[102]Zhu, X., Xiong, Y., Dai, J., Yuan, L., Wei, Y., "Deep feature flow for video recognition", in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 2349–2358.

[103]Carreira, J., Zisserman, A., "Quo vadis, action recognition? a new model and the kinetics dataset", in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017.

[104]Xie, S., Sun, C., Huang, J., Tu, Z., Murphy, K., "Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification", in Proceedings of the European Conference on Computer Vision (ECCV), September 2018.

[105]Beery, S., Wu, G., Rathod, V., Votel, R., Huang, J., "Context r-cnn: Long term temporal context for per-camera object detection", in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 13 075–13 085.

[106]Bertasius, G., Torresani, L., Shi, J., "Object detection in video with spatiotemporal sampling networks", in Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 331–346.

[107]Jaeger, P. F., Kohl, S. A., Bickelhaupt, S., Isensee, F., Kuder, T. A., Schlemmer, H.-P., Maier-Hein, K. H., "Retina u-net: Embarrassingly simple exploitation of segmentation supervision for medical object detection", in Machine Learning for Health Workshop. PMLR, 2020, pp. 171–183.

[108]Roth, H. R., Lu, L., Liu, J., Yao, J., Seff, A., Cherry, K., Kim, L., Summers, R. M., "Improving computer-aided detection using convolutional neural networks and random view aggregation", IEEE Transactions on Medical Imaging, Vol. 35, No. 5, 2016, pp. 1170-1181.

[109]Tang, H., Kim, D. R., Xie, X., "Automated pulmonary nodule detection using 3d deep convolutional neural networks", in 2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018), 2018, pp. 523-526.

[110]Yan, K., Bagheri, M., Summers, R. M., "3d context enhanced region-based convolutional neural network for end-to-end lesion detection", in International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, 2018, pp. 511–519.

[111]Ziabari, A., Rose, D. C., Eicholtz, M. R., Solecki, D. J., Shirinifard, A., "A 2.5d yolo-based fusion algorithm for 3d localization of cells", in 2019 53rd Asilomar Conference on Signals, Systems, and Computers, 2019, pp. 2185-2190.

[112]de Vos, B. D., Wolterink, J. M., de Jong, P. A., Leiner, T., Viergever, M. A., Išgum, I., "Convnet-based localization of anatomical structures in 3-d medical images", IEEE Transactions on Medical Imaging, Vol. 36, No. 7, 2017, pp. 1470-1481.

[113]Lee, S.-g., Bae, J. S., Kim, H., Kim, J. H., Yoon, S., "Liver lesion detection from weakly-labeled multi-phase ct volumes with a grouped single shot multibox detector", in International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, 2018, pp. 693–701.

[114]Ye, J., Ito, S., Toyama, N., "Computerized ultrasonic imaging inspection: From shallow to deep learning", Sensors, Vol. 18, No. 11, Nov 2018, p. 3820, available at: https://doi.org/10.3390/s18113820

[115]Ye, J., Toyama, N., "Benchmarking deep learning models for automatic ultrasonic imaging inspection", IEEE Access, Vol. 9, 2021, pp. 36 986-36 994.

[116]Rentala, V. K., Kanzler, D., Fuchs, P., "Pod evaluation: The key performance indicator for nde 4.0", Journal of Nondestructive Evaluation, Vol. 41, No. 20, 2022.

[117]Virkkunen, I., "Probability of detection (pod)", Webinar, available at: https://trueflaw.com/pod/podintro 2020 (Accessed on 25.01.2022).

[118]Razavian, A. S., Azizpour, H., Sullivan, J., Carlsson, S., "Cnn features off-the-shelf: An astounding baseline for recognition", in 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2014, pp. 512–519.

[119]Shorten, C., Khoshgoftaar, T. M., "A survey on Image Data Augmentation for Deep Learning", Journal of Big Data, Vol. 6, No. 1, Jul. 2019, p. 60, available at: https://doi.org/10.1186/s40537-019-0197-0

Publications

Publication 1

D. Medak, L. Posilović, M. Subašić, M. Budimir, S. Lončarić, ”Automated Defect Detection From Ultrasonic Images Using Deep Learning”, *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 68, no. 10, Oct. 2021, pp. 3126-3134.

Automated Defect Detection From Ultrasonic Images Using Deep Learning

Duje Medak, Luka Posilović, Marko Subašić, *Member, IEEE*, Marko Budimir, *Member, IEEE*, and Sven Lončarić, *Member, IEEE*

***Abstract*— Nondestructive evaluation (NDE) is a set of techniques used for material inspection and defect detection without causing damage to the inspected component. One of the commonly used nondestructive techniques is called ultrasonic inspection. The acquisition of ultrasonic data was mostly automated in recent years, but the analysis of the collected data is still performed manually. This process is thus very expensive, inconsistent, and prone to human errors. An automated system would significantly increase the efficiency of analysis, but the methods presented so far fail to generalize well on new cases and are not used in real-life inspection. Many of the similar data analysis problems were recently tackled by deep learning methods. This approach outperforms classical methods but requires lots of training data, which is difficult to obtain in the NDE domain. In this work, we train a deep learning architecture EfficientDet to automatically detect defects from ultrasonic images. We showed how some of the hyperparameters can be tweaked in order to improve the detection of defects with extreme aspect ratios that are common in ultrasonic images. The proposed object detector was trained on the largest dataset of ultrasonic images that was so far seen in the literature. In order to collect the dataset, six steel blocks containing 68 defects were scanned with a phased-array probe. More than 4000 VC-B-scans were acquired and used for training and evaluation of EfficientDet. The proposed model achieved 89.6% of mean average precision (mAP) during fivefold cross validation, which is a significant improvement compared to some similar methods that were previously used for this task. A detailed performance overview for each of the folds revealed that EfficientDet-D0 successfully detects all of the defects present in the inspected material.**

***Index Terms*— Automated defect detection, deep learning, flaw detection, ultrasonic image analysis, ultrasonic testing (UT).**

I. Introduction

NONDESTRUCTIVE evaluation (NDE) is a set of techniques used for material evaluation and defect detection in industry and science [1]. These methods do not damage the inspected material, which makes them perfect for continuous monitoring of critical components of some systems. NDE methods are used in aeronautics, oil and gas industry, various power plants, and other industries where it is crucial to detect material flaws in time in order to prevent further damages and disasters. A variety of NDE methods are used: ultrasonic, eddy current, thermography, and X-radiography, to name a few. Some of the advantages when using ultrasonic testing (UT) include simple usage, precise extraction of the defect location [2], and the ability to evaluate the structure of alloys of components with different acoustic properties [3]. UT employs a diverse set of methods based on the generation and detection of mechanical vibrations or waves within test objects [4]. One of the commonly used types of probes is called a phased-array probe. A phased-array probe is a multichannel ultrasonic system, which uses the principle of a time-delayed triggering of the transmitting transducer elements, combined with a time corrected receiving of detected signals [5]. Using a phased-array probe increases the reliability of inspections since the material is inspected from various angles. During an inspection, a probe is moved along the surface of the inspected component. At each position, the probe transmits and receives ultrasound energy. The amount of received energy is usually shown as a function of time in the representation called A-scan. Multiple A-scans are obtained when the probe is moved along one axis. The sequence of A-scans can then be visualized as an image called B-scan. Each column from the B-scan is obtained from the A-scan by converting the amplitude at a specific point in time into pixel intensity. The dimension of the inspected component and the resolution of the inspection determine the width of the B-scan. It is common to see B-scans that were created from several hundreds of A-scans. Other representations of UT data, such as volume corrected B-scans (VC-B-scans), C-scans, and D-scans, are also often used during the analysis. Some of the mentioned representations are shown in Fig. 1.

The acquisition of the UT data was mostly automated in recent years, but the analysis of the acquired data is still performed manually by trained experts. The amount of data that needs to be analyzed is immense, especially when a phased-array probe is used. The success of the analysis depends solely on the analyzer's knowledge and experience making this process prone to errors. Many efforts were made in order to develop methods that could assist the analyzers in the defect detection process.

The most popular approach for the automated analysis of ultrasonic data in NDE is based on the wavelet transform

Manuscript received January 20, 2021; accepted May 17, 2021. Date of publication May 19, 2021; date of current version September 27, 2021. This work was supported by the European Union through the European Regional Development Fund under Grant (Smart UTX) KK.01.2.1.01.0151. *(Corresponding author: Duje Medak.)*

Duje Medak, Luka Posilović, Marko Subašić, and Sven Lončarić are with the Faculty of Electrical Engineering and Computing, University of Zagreb, 10000 Zagreb, Croatia (e-mail: duje.medak@fer.hr).

Marko Budimir is with the INETEC Ltd., 10000 Zagreb, Croatia.

Digital Object Identifier 10.1109/TUFFC.2021.3081750

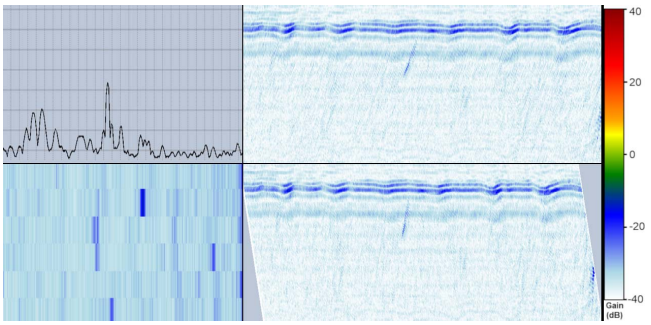


Fig. 1. Examples of different ultrasonic data representations. Top left: A-scan. Beneath it is the C-scan representation. Right: B-scan (top) and volume corrected B-scan (bottom).

of A-scans. Coefficients from the transformation can then be used as an input to some classifiers, such as artificial neural networks (ANNs) [6], [7] or support vector machines (SVMs) [8]–[11]. This approach works well when the dataset is limited since the feature extraction is predefined and the data samples are used solely for classifier training. Different kinds of transformations, such as Fourier transformation or Cosine transformation, can also be used in the feature extraction step as shown in [12]. Usage of a special type of neural network called convolutional neural network (CNN) is becoming more popular in recent years for the analysis of sequences and grid-like representations of the data. Some works [13], [14] showed that methods based on CNN achieve good results when applied for UT data analysis. While many authors achieved good results, datasets that were used for evaluation contained only a few thousand or even a few hundreds of A-scans. When an inspection is performed in a real-life situation, millions of A-scans are usually acquired. A small dataset containing only a fraction of that amount can hardly capture all the possible appearance variations of the signal. The main drawback of A-scan analysis is the lack of context from the surrounding area. Distinguishing between geometry, noise, and defect signals would be considerably easier if the information from the surrounding A-scans would be available.

This problem is solved if B-scans are used for the automated analysis. In this case, the spatial information from the surrounding area is available. This extra information can be used to improve defect detection while reducing the number of false positives (FPs). The wavelet transform that was commonly used for A-scan analysis also proved to be a useful tool when dealing with images. Cygan *et al.* [15] first denoised images using the wavelet transform and then performed defect detection using the Radon transform. In [16], the wavelet transform was used for feature extraction. The authors also tried Gabor filters but determined that the wavelet transform achieves better results. The authors of that work used Fuzzy C-Mean clustering to classify extracted features. With the development of deep learning models, new approaches for the image analysis of the UT data based on CNNs appeared. Ye *et al.* [2] compared a CNN with the traditional approaches that use handcrafted descriptors in combination with SVM. The authors demonstrated that the CNN-based approach yields superior results. Some recent works [17]–[19] also showed that CNN architectures can successfully be applied to analyze the UT data. Virkkunen *et al.* [17] showed that a custom CNN can be trained on artificially generated data. The performance of this approach was compared with the human expert's performance of detection and the authors concluded that automated analysis using the deep learning approach works better and has a higher probability of detection (POD). However, this approach was not tested on a real (nongenerated) dataset of B-scans. In [18], it was demonstrated how the data needed for the training of a deep learning architecture can be simulated. The authors used simulated data to train a network for crack characterization. The proposed deep learning approach was compared to the 6-dB drop method. The deep learning model was able to size 97% of the tested defects of lengths 1–5 mm within ± 1 mm, while the 6 dB method could only size 48% of the defects. In [19], real B-scan data were used to train popular object detectors. To deal with a limited amount of data, the authors used the pretrained architectures YOLOv3 [20] and SSD [21] and performed heavy data augmentation during the training. Even though the achieved results were good, the amount of testing data was very small (only 98 images).

In this work, we perform defect detection with EfficientDet architecture, a state-of-the-art object detection algorithm that was not used for this problem before. EfficientDet belongs to the one-stage family of detectors, meaning that the objects are searched in the predefined rectangles called anchors or default boxes [21]. Anchors are rough guesses about the objects' dimensions and positions in the image. The shapes and positions of the anchors are determined from the hyperparameters provided during the training. We propose a novel procedure for calculating anchors' hyperparameters values in order to improve the detection of defects with extreme aspect ratios that are common in UT images. To the best of our knowledge, detection algorithms with the ability of defect localization from UT B-scan were only previously shown in [19]. We compared EfficientDet with the best performing method from that work, YOLOv3. In addition, we also made a comparison with the RetinaNet, an improved version of the other method used in that work, SSD. We showed that some previous works used CNNs for defect detection, but the approach proposed in this work has the following merits.

1) We are the first to employ a state-of-the-art object detector EfficientDet on this task. We proposed a novel procedure for calculating anchors' hyperparameters and demonstrated that using the calculated values improves the model's average precision by a significant amount.
2) We used the largest dataset of real UT B-scans for training and evaluation that was used so far in the literature (over 4000 images). The collected database displays 68 unique defects that were created using various methods. This ensures that the obtained results represent a realistic performance of the proposed defect detection method.
3) We divide our dataset into five disjunctive subsets (folds) and perform fivefold cross validation [22]. We then conduct a detailed analysis of model performance for each of the folds. This is used to prove that the proposed architecture is reliable enough to be used for automated

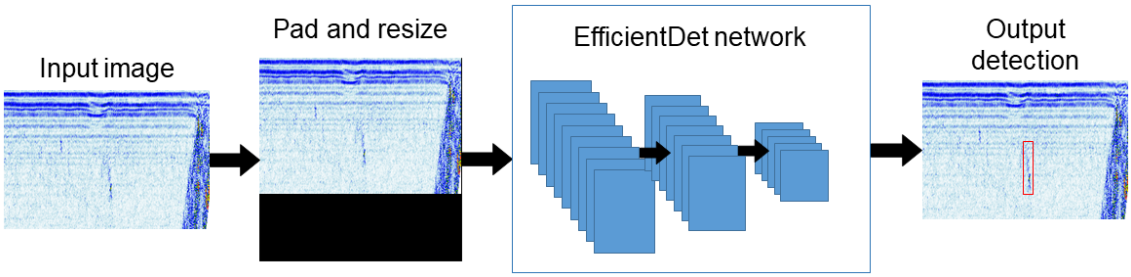


Fig. 2. Illustration of the proposed approach. The input image is first padded to get a squared image and then resized to the network's input size. The preprocessed image is fed to trained EfficientDet [23] architecture. Finally, nonmaximum suppression and confidence thresholding are applied to the model's output to ensure that only relevant detections are kept.

defect detection. Performing such rigorous testing is the most thorough evaluation that was so far done to test the performance of a deep learning object detector for defect detection from ultrasonic images.

II. Materials and Methods

Fig. 2 shows a high-level illustration of the proposed method. A VC-B-scan is padded to get an image of equal width and height. The image is resized to 512×512 pixels and fed into the EfficientDet object detection algorithm. The output of the network is a list of bounding boxes and associated confidences. Nonmaximum suppression is performed to ensure that duplicate boxes are removed. Finally, by applying the confidence threshold, we only keep the boxes with higher probabilities.

A. Used Deep Learning Architecture

Deep learning object detectors can be divided into two categories: one-stage detectors, such as YOLOv1 [24], SSD [21], YOLOv2 [25], RetinaNet [26], and YOLOv3 [20], and two-stage detectors, such as R-CNN [27], Fast-RCNN [28], and Faster-RCNN [29]. One-stage detectors search for the object's presence at predefined positions. This is usually implemented as a dense grid of rectangular-shaped areas where the model decides for each area whether it contains some particular object or not. Two-stage detectors first run a region proposal algorithm and then classify only the proposed areas. The number of areas that need to be classified is decreased, but the overall complexity is increased because an extra step for region proposal is needed. Two-stage detectors are usually slower but more accurate compared to one-stage detectors, but this accuracy gap was recently reduced. Reliability of defect detection should always be the primary criterion for choosing the proper model, but considering the amount of data that needs to be analyzed, it would be beneficial if the used model was fast. A good tradeoff between accuracy and speed is offered by the EfficientDet model [23], which belongs to the one-stage detector family.

This architecture was developed with computational efficiency in mind. The authors created a base model (EfficientDet-D0) that can be scaled up depending on the available resources. The family of EfficientDet models includes a total of eight models (D0–D7). However, having a more complex network does not always lead to an improvement, especially if the objects are simple like it is in the case with defects from the UT data. Experimental results that we presented in Section IV show that the performance of the smaller and faster EfficientDet-D0 model is better than those of EfficientDet-D1 and EfficientDet-D2. Like other one-stage detectors, EfficientDet searches for object presence at predefined areas called anchors (default boxes). This dense grid of rectangles covers a variety of different shapes. Every anchor-based object detector needs a list of aspect ratios and scales as an input to calculate the shape and size of anchors. Object detectors predict the locations of the objects with respect to these anchors. Having proper anchors hyperparameters can speed up model training and improve accuracy. There are several approaches [30]–[32] that can be used to estimate good anchors hyperparameters, but it is often needed to make some assumptions about the objects' shapes so that the calculation would be possible. When working with natural images, most of the researchers simply use default values of aspect ratios and scales for RetinaNet or EfficientDet. These values are shown in Table I. Looking at Fig. 3, one can notice that the aspect ratios of our bounding boxes are much more extreme. We decided to calculate new values using K-means with the Jaccard distance, as proposed in YOLOv2 [25]. Values obtained from this procedure can be used directly for training the YOLOv3 model, but using them to train RetinaNet or EfficientDet is not straightforward. Aspect ratios and scales for RetinaNet and EfficientDet were calculated in the following way.

TABLE I
Comparison Between Commonly Used Default Values of Aspect Ratios and Scales With the Values We Used in This Work

	default	ours
aspect ratios	[0.5, 1, 2]	[3.77 5.73 6.94 10.05 13.56]
scales	$[2^{0/3}, 2^{1/3}, 2^{2/3}]$	[0.88 1.34 1.46] *
*scales before merging [1.34, 1.49, 1.43, 0.88, 0.89]		

TABLE II
DATASET OVERVIEW

	number of defects	number of images	number of annotations
fold 1	14	1006	1317
fold 2	15	915	1439
fold 3	16	872	1437
fold 4	12	298	1316
fold 5	11	1083	1128
total	68	4174	6637

1) First, we used K-means with the Jaccard distance to calculate five default shapes. This procedure calculates which shapes of bounding boxes will on average fit the best to the samples (after they are resized to the input image size) from the dataset. The obtained widths and heights are expressed as absolute values (in pixels), so they need to be converted into a list of aspect ratios and scales.
2) Aspect ratios can simply be calculated by dividing the height of each shape by its width.
3) Determining scale requires knowledge about the detection process of EfficientDet. EfficientDet performs object detection on five different scales. In order for that to be possible, five feature maps (called P3–P7 in the original publication [23]) of different resolutions are used. Each of these feature maps has an assigned template anchor size. Sizes of template anchors range from 32×32 for the P3 feature map to 512×512 for the P7 feature map. We calculated the scales by finding which of the template anchors is the most similar to our shape. As a similarity measure, we used the absolute difference between template anchor size and the bigger side of our calculated shape. Once we know which template anchor is the most similar, we can calculate the scale factor by dividing the maximum size of our shape by the anchor size. The described procedure for scales calculation can be written mathematically as shown in 1a and 1b.
4) To decrease the total number of anchors, we merged the values of scales that were similar.

The final values are shown in Table I. It can be seen that the calculated values greatly differ from the commonly used default values. In Section IV, we proved that using these values improves the performance of the EffcientDet model by a significant amount

$$s_i = \frac{\max(\text{width}(BB_i), \text{height}(BB_i))}{BTA_i} \tag{1a}$$

$$BTA_i = \arg\min_{T_j} |\max(\text{width}(BB_i), \text{height}(BB_i)) - T_j)|$$
$$T_j \in \{32, 64, 128, 256, 512\} \tag{1b}$$

where

BB_i ith shape calculated using K-means.
T_j template anchor size.

B. Dataset

For the development and evaluation of the proposed method, we used an in-house dataset. Creating test specimens with artificial defects inside is a costly process. Companies have to invest a fair amount of money for the acquisition of those blocks as well as the equipment that is needed to perform UT. It is only logical that they would like to keep that data private in order to maintain their competitiveness compared to other companies. Besides the dataset provided in [17], which was artificially generated, there are no publicly available datasets that could be used for the development and evaluation of methods for UT analysis. Having a large dataset ensures the credibility of the obtained results. It also makes training of a large deep learning model (with tens of millions of parameters) possible. The data used in this work was obtained by scanning six stainless steel blocks. Blocks contained between six and 34 defects. Defects were artificially created using various methods leading to different types of defects, such as side-drilled holes, flat bottom holes, thermal fatigue cracks, mechanical fatigue cracks, electric discharge machined notches, solidification cracks, and incomplete penetration of the weld. The scanning was done using the INETEC Dolphin scanner in combination with INETEC dual-phased-array probe with 2×16 elements, element dimensions 1.45 mm (pitch) $\times$ 1.3 mm (width), longitudinal wave, the central frequency of 2.25 MHz, and the frequency average bandwidth $\geq$70% at $-$6-dB gain. The collected data include only the shallow parts of the blocks (up to 200 mm). After all of the data were collected, multiple human experts analyzed it and determined the positions of the defects. The location of each defect was annotated by a bounding box. Even though all of the positions of the defects were known from the blocks' blueprints, manual annotation is needed to ensure that only the visible defects will be labeled. One of the most important questions about the experimental setup is how to split the data. Using a hold-out method is the most common way to test the performance of the model, but it is not the most reliable. Even if unique images are contained in the test set, they often represent some defects that already appeared in the train set (but on the image from a different angle for example). Having similar images in the train and test subset would lead to an unrealistically good model's performance. To provide a fair evaluation, we decided to split all of our data into five subsets (folds) where each fold contains unique defects, as shown in Table II. This ensures that all of the images used for testing as well as the defects that are displayed in those images are unique and will not be used for training. Each fold was made to contain approximately 20% of all available annotations. The width of most of the original images is a lot larger than their height. This can cause problems since some

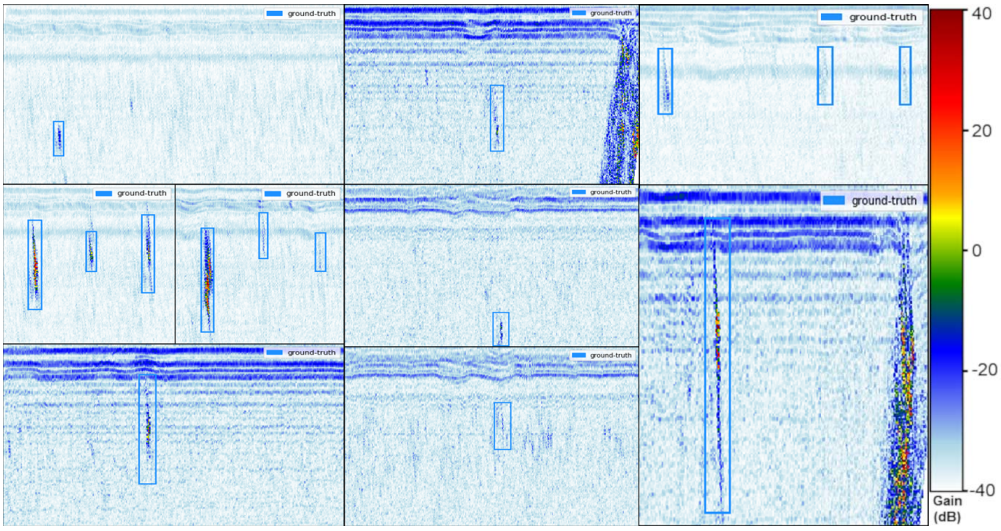


Fig. 3. Examples of the used VC-B-scans with ground-truth labels.

defects would not be easily seen after padding and resizing the image. To avoid this, we split some of the images into multiple patches. The resulting images have an aspect ratio closer to 1, so the amount of needed padding is minimized. The height of the images varies between 200 and 375 pixels, while their width varies between 300 and 400 pixels. Defects displayed in a B-scan usually appear slanted, so the bounding boxes do not fit perfectly around them. Since each of the acquired A-scans was taken at some angle and projected into exactly one image column, B-scans do not display the internal structure of the material realistically. If VC-B-scan is used, each A-scan is transferred onto the image at the same angle that the ultrasonic waves were propagated through the material. This skews VC-B-scans as shown in Fig. 1, but the orientations of the displayed defects are more similar to the physical orientation of the defects inside of the material. Even though image representation of UT data is naturally in the grayscale colormap, B-scans are often colored for easier manual inspection. We also used pseudo-colored images that were exported using the INETEC SignyOne ultrasound data acquisition and analysis software. A few example images from the dataset are shown in Fig. 3.

III. Experimental Setup

A. Model Training

We trained three representatives from the EfficientDet family (EfficientDet-D0, EfficientDet-D1, and EfficientDet-D2). We tried two approaches for weight initialization as follows:

1) randomly initialized weights;
2) weights from a model pretrained on COCO [33] dataset.

Using cross validation to evaluate the performance of the model means that every model is trained five times. Each time a different fold is left out as a test set, while the four remaining folds are used to train the model. In addition, we also left out 15% of the training subset for validation. The validation subset was used to decrease the learning rate on plateaus and early stopping of the training. The training subset was augmented during the training, which is commonly done to improve the generalization of the model and increase precision. Following transformations were used: horizontal flip, random crop, translation, and visual effects (contrast, brightness, and color enhancement). We trained EfficientDet-D0 with batch size 8 and 500 steps per epoch. EfficientDet-D1 and EfficientDet-D2 were trained with batch size 4 and 1000 steps per epoch. All of the models were trained using the Adam optimizer with an initial learning rate of $1e^{-3}$. The training was performed on a single NVIDIA RTX 2080 Ti GPU on a machine with AMD Threadripper 1920X and 128 GB of RAM. We compared the performance of the EfficientDet model with two popular object detectors YOLOv3 and RetinaNet (with ResNet [34] backbone). The same hyperparameters (optimizer, batch size, number of steps, and callback hyperparameters) as the ones used for the EfficientDet-D0 model were used when training these models. To have a fair comparison, these models were also pretrained on the COCO dataset. We calculated anchors for YOLO using K-means as described in [25]. For RetinaNet, we used the same values as for EfficientDet (described in Section II).

B. Evaluation Metric

The mean average precision (mAP) metric as given in the later versions of PASCAL VOC (2010–2012) [35] was used as an evaluation metric. This is a common metric to compare the performance of object detectors. The value of mAP is determined by the area under the precision–recall curve. In order to calculate the curve, the number of true positives (TPs), FPs, and false negatives needs to be calculated first. Each output detection of the model contains the coordinates of the bounding box and a probability of that box containing a defect. To determine which output predictions are TPs, the intersection over the union between the predicted bounding boxes and the ground-truth labels needs to be calculated. IOU

TABLE III
MEAN AVERAGE PRECISION FOR DIFFERENT CONFIGURATIONS OF EFFICIENTDET-D0 MODEL

model	mAP
EfficientDet-D0 (512x512), default anchors, pretrained on coco	0.837
EfficientDet-D0 (512x512), custom anchors, pretrained on coco	**0.896**
EfficientDet-D0 (512x512), custom anchors, random weights initialization	0.875
EfficientDet-D0 (384x384), custom anchors, pretrained on coco	0.881

TABLE IV
MEAN AVERAGE PRECISION FOR EACH OF THE FOLDS. BOLD TEXT INDICATES THE BEST PERFORMANCE FOR THAT FOLD

model	fold1	fold2	fold3	fold4	fold5	average
YOLOv3 (416x416)	0.846	0.787	0.756	0.901	0.742	0.806
RetinaNet (ResNet50)	0.856	**0.833**	0.826	0.924	0.832	0.854
RetinaNet (ResNet101)	0.829	0.831	0.818	0.920	0.794	0.838
RetinaNet (ResNet152)	0.872	0.821	0.830	0.901	0.850	0.855
EfficientDet-D0	**0.937**	0.829	**0.879**	**0.943**	0.893	**0.896**
EfficientDet-D1	0.927	0.793	0.869	0.917	**0.901**	0.881
EfficientDet-D2	0.936	0.780	0.826	0.920	0.895	0.871

is calculated as shown in the following equation:

$$\text{iou} = \frac{\text{area}(BB_{\text{pred}} \cap BB_{\text{gt}})}{\text{area}(BB_{\text{pred}} \cup BB_{\text{gt}})} \tag{2}$$

where

BB_{pred} predicted bounding box.
BB_{gt} ground-truth bounding box.

Predicted bounding boxes that have an intersection over union (IOU) with some ground-truth label higher than 0.5 are considered TPs. Predicted bounding boxes that do not have matching ground-truth boxes are considered FPs, and the ground-truth boxes that were not matched with any predicted bounding box are considered false negatives (FN). The numbers of TPs, FPs, and false negatives are then used to calculate precision and recall as shown in 3a and 3b

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{3a}$$

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3b}$$

where

TP number of true positive predictions.
FN number of false negative predictions.
FP number of false positive predictions.

By changing the confidence threshold, we can get precision values for different recall values and plot the precision–recall curve. The area under that curve is used to compare the performances of different models.

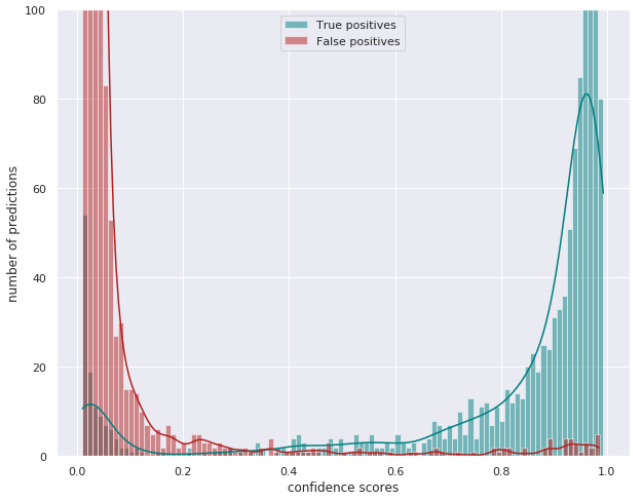


Fig. 4. Histogram of EfficientDet-D0 confidence scores and the density estimate lines using a Gaussian kernel. Nonmaximum suppression with a threshold of 0.3 was done before plotting.

IV. RESULTS AND DISCUSSION

In order to determine the best configuration for EfficientDet, we run a few experiments with different setups. Some of our findings can be seen in Table III. We showed that calculating aspect ratios and scales as proposed in this work improves the mAP by almost 6%. We also showed that using a smaller input image resolution (384 × 384 pixel) decreases the model's performance even though most images from our dataset are smaller than 384 × 384 pixel. We think that this has to do with the architecture of EfficientNet that downsamples the input image in an early stage, which leads to information loss. Comparison of EfficientDet with YOLOv3 and RetinaNet is shown in Table IV. We experimentally determined that RetinaNet performs better if the input images are only padded, so we did not resize the images as we did for EfficientDet and YOLOv3. Even the smallest baseline model EfficientDet-D0, which performs worse than RetinaNet on common benchmark datasets such as COCO [33] and PASCAL [35], outperformed the best version of RetinaNet by

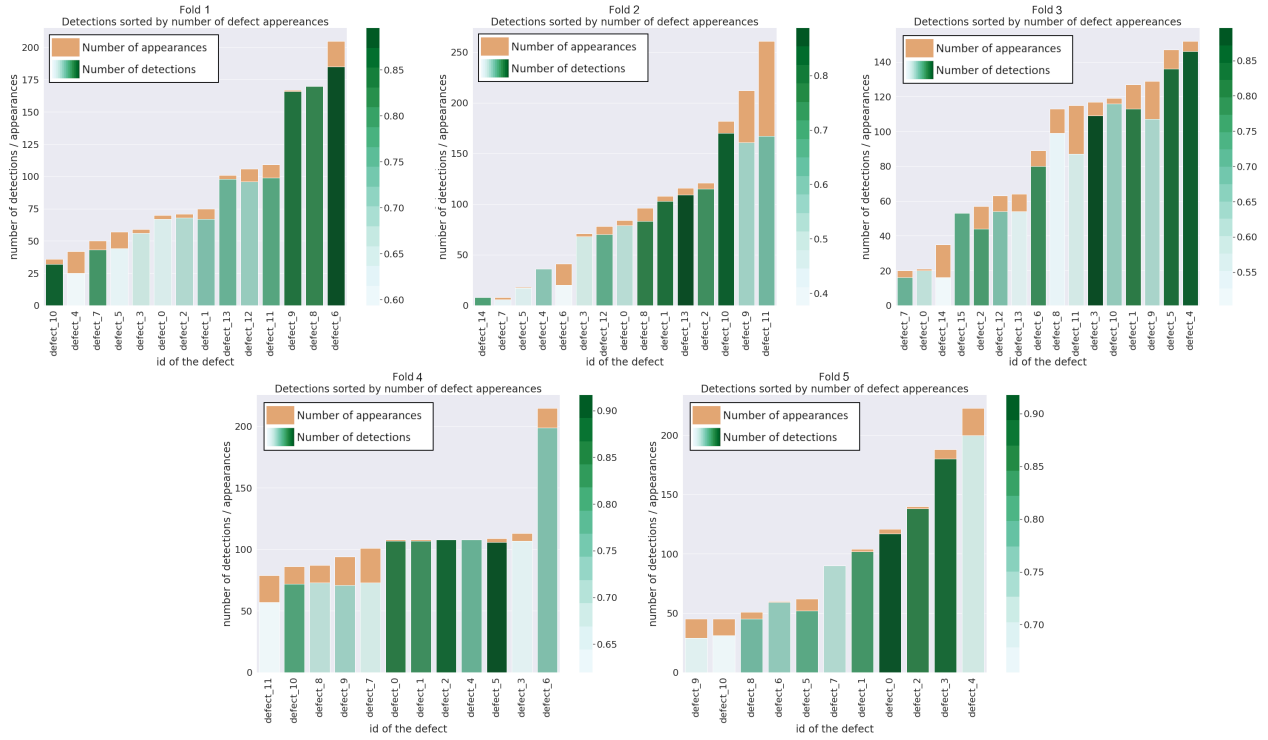


Fig. 5. Detection results obtained with EfficientDet-D0 with confidence threshold 0.3. Each of the bars shows how many appearances of the defect did the model detect. The color hue of the bar represents the maximum confidence from all of the detection for that defect. Best viewed in color.

more than 4%. EfficientDet-D1 and EfficientDet-D2 perform worse than the baseline EfficientDet-D0 model. Besides being the most accurate from all of the tested models, the smallest architecture EfficientDet-D0 is also the fastest. The average inference time of the used EfficientDet-D0 model on NVIDIA RTX 2080 Ti GPU is 26 ms. We determined from the obtained results that the EfficientDet-D0 is the most suitable choice for automated defect detection. We did not make any modifications to the EfficientDet-D0 architecture that is specific to the used hardware configuration, so we believe that this architecture can also be applied for similar tasks in the other NDE technologies. Due to the small number of parameters that have to be trained, this architecture is also very convenient for situations in which the number of available images is limited.

When the inference on the new data is performed, a confidence threshold needs to be set to limit the number of FPs. In Fig. 4, we showed how the confidences of the chosen EfficientDet-D0 model relate to the number of FPs and TPs. The Gaussian kernel density estimate lines are also shown in the figure. This plot was calculated for one specific fold, but similar distributions are obtained for other folds as well. Each prediction that has an IOU overlap with the ground-truth annotation greater than 0.5 was considered TPs. This definition causes a small increase of the TPs for the small confidence threshold values because several predictions are matched with a ground-truth label. The confidence threshold is usually set to 0.5, but looking at Fig. 4, we noticed that we could set the threshold to a lower value without significantly increasing the number of FPs. We set it to 0.3 because it is roughly the value for which the ratio of TPs and FPs becomes greater than one. Even if the confidence threshold of 0.3 is used, some of the TPs will be removed, so it is important to test whether the proposed model is able to detect all of the defects. To determine this, we performed a detailed analysis for each of the test folds. In Fig. 5, we showed exactly how many times some defect from the test fold can be seen (how many annotations of the same defect we have). We also showed the number of detections when using the EfficientDet-D0 model with a threshold of 0.3. The proposed model successfully detected 87.5% of the annotations. However, it is important to note that all of the defects have at least one detection, meaning that none of the defects will pass undetected. In fact, the EfficientDet-D0 detected on average 85.7% of appearances of some defect. Undetected annotations are usually some borderline cases for which the defect's signal becomes too weak and even the human operators would not annotate it if they did not confirm their decision by looking at the block's blueprints. The percentage of FPs when using a threshold of 0.3 is 16.7%. We think that this could be decreased by converting the predicted bounding boxes into real-life coordinates and performing some postprocessing/filtering. To compare the results with the previous state of the art, we performed the same detailed analysis for the YOLOv3 model. We set the object threshold to 0.3 even though this value is too low for YOLOv3 architecture and causes a large number of FPs (almost 50%). Even with such a low threshold, this model was not able to detect all of the defects. There were two defects from the fold3 (defect 7 and defect 2) and two defects from the fold5 (defects 9 and 10) for which the YOLOv3 did not manage to detect any annotations. Finally, we also tested the RetinaNet model with

a ResNet152 feature extractor. We again used a confidence threshold of 0.3, which resulted in 19% of false-positive predictions. This model was also unable to detect all of the defects. Defect 4 from the fold1 and defects 7 and 6 from the fold2 did not have any detections. The presented results show that using the proposed EfficientDet-D0 model not only improves the mAP but also enables the detection of all of the defects in the material.

V. Conclusion

Manual analysis of the UT data is a time-consuming and laborious process prone to human error. In order to automate this process and help human experts with the analysis, a reliable method must be developed. In this work, we demonstrated that the EfficientDet-D0 architecture can successfully be adapted to detect defects from images obtained with a phased-array probe. We proposed a novel procedure for calculating the anchors' hyperparameters and showed that this increases the performance of the network significantly. The proposed EfficientDet-D0 model achieved an mAP of 89.6%, which is an improvement of 9% compared to the previous state-of-the-art architecture YOLOv3. While the presented results prove that the EfficientDet-D0 successfully detects all of the defects from the material, it would be useful to compare its performance to the performance of human inspectors. This can be done by performing a POD study, but such study goes beyond the scope of this work. If proven to be equally reliable as the human inspectors, methods similar to the one presented in this work could soon be used in real-life situations in order to assist the human operators with the analysis of the UT data.

References

[1] L. Cartz, *Nondestructive Testing: Radiography, Ultrasonics, Liquid Penetrant, Magnetic Particle, Eddy Current*. Materials Park, OH, USA: ASM International, 1995. [Online]. Available: https://books.google.hr/books?id=0spRAAAAMAAJ

[2] J. Ye, S. Ito, and N. Toyama, "Computerized ultrasonic imaging inspection: From shallow to deep learning," *Sensors*, vol. 18, no. 11, p. 3820, Nov. 2018, doi: 10.3390/s18113820.

[3] S. Davì *et al.*, "Correction of B-scan distortion for optimum ultrasonic imaging of backwalls with complex geometries," *Insight, J. Brit. Inst. Non-Destructive Test.*, vol. 62, no. 4, pp. 184–191, Apr. 2020.

[4] D. Forsyth, "Nondestructive testing of corrosion in the aerospace industry," in *Corrosion Control in the Aerospace Industry* (Woodhead Publishing Series in Metals and Surface Engineering), S. Benavides, Ed. Cambridge, U.K.: Woodhead Publishing, 2009, ch. 5, pp. 111–130. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9781845693459500050

[5] L. von Bernus, A. Bulavinov, D. Joneit, M. Kröning, M. Dalichov, and K. M. Reddy, "Sampling phased array: A new technique for signal processing and ultrasonic imaging," in *Proc. Eur. Conf. Non-Destructive Test. (ECNDT)*, Berlin, Germany, 2006. [Online]. Available: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.218.3412&rep=rep1&type=pdf

[6] F. Bettayeb, T. Rachedi, and H. Benbartaoui, "An improved automated ultrasonic nde system by wavelet and neuron networks," *Ultrasonics*, vol. 42, no. 1, pp. 853–858, 2004, doi: 10.1016/j.ultras.2004.01.064.

[7] S. Sambath, P. Nagaraj, and N. Selvakumar, "Automatic defect classification in ultrasonic NDT using artificial intelligence," *J. Nondestruct. Eval.*, vol. 30, no. 1, pp. 20–28, Mar. 2011, doi: 10.1007/s10921-010-0086-0.

[8] M. Khelil, M. Boudraa, A. Kechida, and R. Drai, "Classification of defects by the SVM method and the principal component analysis (PCA)," *Int. J. Electr. Comput. Eng.*, vol. 1, no. 9, pp. 1–6, 2007, doi: 10.5281/zenodo.1060751.

[9] V. Matz, M. Kreidl, and R. Smid, "Classification of ultrasonic signals," *Int. J. Mater. Product Technol.*, vol. 27, no. 3/4, pp. 145–155, 2006, doi: 10.1504/IJMPT.2006.011267.

[10] A. Al-Ataby, W. Al-Nuaimy, C. R. Brett, and O. Zahran, "Automatic detection and classification of weld flaws in TOFD data using wavelet transform and support vector machines," *Insight, Non-Destructive Test. Condition Monitor.*, vol. 52, no. 11, pp. 597–602, Nov. 2010, doi: 10.1784/insi.2010.52.11.597.

[11] Y. Chen, H.-W. Ma, and G.-M. Zhang, "A support vector machine approach for classification of welding defects from ultrasonic signals," *Nondestruct. Test. Eval.*, vol. 29, no. 3, pp. 243–254, Jul. 2014, doi: 10.1080/10589759.2014.914210.

[12] F. C. Cruz, E. F. S. Filho, M. C. S. Albuquerque, I. C. Silva, C. T. T. Farias, and L. L. Gouvêa, "Efficient feature selection for neural network based detection of flaws in steel welded joints using ultrasound testing," *Ultrasonics*, vol. 73, pp. 1–8, Jan. 2017, doi: 10.1016/j.ultras.2016.08.017.

[13] M. Meng, Y. J. Chua, E. Wouterson, and C. P. K. Ong, "Ultrasonic signal classification and imaging system for composite materials via deep convolutional neural networks," *Neurocomputing*, vol. 257, pp. 128–135, Sep. 2017, doi: 10.1016/j.neucom.2016.11.066.

[14] N. Munir, H.-J. Kim, J. Park, S.-J. Song, and S.-S. Kang, "Convolutional neural network for ultrasonic weldment flaw classification in noisy conditions," *Ultrasonics*, vol. 94, pp. 74–81, Apr. 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0041624X18305754

[15] H. Cygan, L. Girardi, P. Aknin, and P. Simard, "B-scan ultrasonic image analysis for internal rail defect detection," in *Proc. World Congr. Railway Res.*, Oct. 2003, pp. 1–6.

[16] A. Kechida, R. Drai, and A. Guessoum, "Texture analysis for flaw detection in ultrasonic images," *J. Nondestruct. Eval.*, vol. 31, no. 2, pp. 108–116, Jun. 2012, doi: 10.1007/s10921-011-0126-4.

[17] I. Virkkunen, T. Koskinen, O. Jessen-Juhler, and J. Rinta-Aho, "Augmented ultrasonic data for machine learning," *J. Nondestruct. Eval.*, vol. 40, no. 1, pp. 1–11, Mar. 2021.

[18] R. J. Pyle, R. L. T. Bevan, R. R. Hughes, R. K. Rachev, A. A. S. Ali, and P. D. Wilcox, "Deep learning for ultrasonic crack characterization in NDE," *IEEE Trans. Ultrason., Ferroelectr., Freq. Control*, vol. 68, no. 5, pp. 1854–1865, May 2021.

[19] L. Posilović, D. Medak, M. Subašić, T. Petković, M. Budimir, and S. Lončarić, "Flaw detection from ultrasonic images using YOLO and SSD," in *Proc. 11th Int. Symp. Image Signal Process. Anal. (ISPA)*, Sep. 2019, pp. 163–168.

[20] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*. [Online]. Available: http://arxiv.org/abs/1804.02767

[21] W. Liu *et al.*, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.*, vol. 9905, Oct. 2016, pp. 21–37.

[22] T. Hastie, J. Friedman, and R. Tisbshirani, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York, NY, USA: Springer, 2017.

[23] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and efficient object detection," 2019, *arXiv:1911.09070*. [Online]. Available: http://arxiv.org/abs/1911.09070

[24] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/CVPR.2016.91

[25] J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger," *CoRR*, vol. abs/1612.08242, pp. 1–9, Dec. 2016. [Online]. Available: http://arxiv.org/abs/1612.08242

[26] T. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *CoRR*, vol. abs/1708.02002, pp. 1–10, Aug. 2017. [Online]. Available: http://arxiv.org/abs/1708.02002

[27] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *CoRR*, vol. abs/1311.2524, pp. 1–21, Nov. 2013. [Online]. Available: http://arxiv.org/abs/1311.2524

[28] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448, doi: 10.1109/ICCV.2015.169.

[29] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/TPAMI.2016.2577031

[30] M. Zlocha, Q. Dou, and B. Glocker, "Improving RetinaNet for CT lesion detection with dense masks from weak RECIST labels," 2019, *arXiv:1906.02283*. [Online]. Available: http://arxiv.org/abs/1906.02283

[31] M. Ahmad, M. Abdullah, and D. Han, "Small object detection in aerial imagery using RetinaNet with anchor optimization," in *Proc. Int. Conf. Electron., Inf., Commun. (ICEIC)*, Jan. 2020, pp. 1–3.

[32] Y. Zhong, J. Wang, J. Peng, and L. Zhang, "Anchor box optimization for object detection," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2020, pp. 1275–1283.
[33] T.-Y. Lin *et al.*, "Microsoft COCO: Common objects in context," in *Computer Vision—ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham, Switzerland: Springer, 2014, pp. 740–755.
[34] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, pp. 1–12, Dec. 2015. [Online]. Available: http://arxiv.org/abs/1512.03385
[35] M. Everingham, L. van Gool, C. Williams, J. Winn, and A. Zisserman. (2012). *The PASCAL Object Recognition Database Collection*. Accessed: May 1, 2020. [Online]. Available: http://host.robots.ox.ac.uk/pascal/VOC/

Duje Medak received the M.Sc. degree from the Faculty of Electrical Engineering and Computing, University of Zagreb, Zagreb, Croatia, in 2019, where he is currently pursuing the Ph.D. degree.

He is currently working as a Researcher with Image Processing Group, Department of Electronic Systems and Information Processing, University of Zagreb. His research interests include image processing, image analysis, machine learning, deep learning, and deep learning object detection methods and their application in the NDE domain.

Luka Posilović received the M.Sc. degree from the Faculty of Electrical Engineering and Computing, University of Zagreb, Zagreb, Croatia, in 2019, where he is currently pursuing the Ph.D. degree.

He is also working as a Young Researcher with Image Processing Group, University of Zagreb. His research interests include visual quality control, object detection, and synthetic image generation.

Marko Subašić (Member, IEEE) received the Ph.D. degree from the Faculty of Electrical Engineering and Computing, University of Zagreb, Zagreb, Croatia, in 2007.

Since 1999, he has been working with the Department for Electronic Systems and Information Processing, Faculty of Electrical Engineering and Computing, University of Zagreb, where he is currently working as an Associate Professor. He teaches several courses at the graduate and undergraduate levels. His research interests include image processing and analysis and neural networks, with a particular interest in image segmentation, detection techniques, and deep learning.

Dr. Subašić is a member of the Croatian Center for Computer Vision, the Croatian Society for Biomedical Engineering and Medical Physics, and the Centre of Research Excellence for Data Science and Advanced Cooperative Systems.

Marko Budimir (Member, IEEE) received the M.Sc. degree in physics from the Faculty of Science, University of Zagreb, Zagreb, Croatia, in 2000, and the Ph.D. degree from the École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland, in 2006.

From 2006 to 2008, he worked at EPFL. Since 2008, he has been working with the Institute of Nuclear Technology (INETEC). He coordinated many key projects at INETEC. Although he is a key person in a company of industry sector, he is still working close to the field of science.

Sven Lončarić (Member, IEEE) received the Ph.D. degree in electrical engineering from the University of Cincinnati, Cincinnati, OH, USA, in 1994, as a Fulbright Scholar.

He is currently a Full Professor of electrical engineering and computer science with the Faculty of Electrical Engineering and Computing, University of Zagreb, Zagreb, Croatia. With his students and collaborators, he has coauthored more than 200 publications in scientific journals and conferences. He is the Founder of the Center for Computer Vision, University of Zagreb, where he is also the Head of the Image Processing Group. He has served as the Co-Director for the National Center of Research Excellence in Data Science and Cooperative Systems.

Prof. Lončarić is a member of the Croatian Academy of Technical Sciences. He received several awards for his scientific and professional work. He was the Chair of the IEEE Croatia Section.

Publication 2

D. Medak, L. Posilović, M. Subašić, M. Budimir, S. Lončarić, ”DefectDet: a deep learning architecture for detection of defects with extreme aspect ratios in ultrasonic images”, *Neurocomputing*, vol. 473, Dec. 2021, pp. 107-115.

Neurocomputing 473 (2022) 107–115

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

DefectDet: A deep learning architecture for detection of defects with extreme aspect ratios in ultrasonic images

Duje Medak [a,*], Luka Posilović [a], Marko Subašić [a], Marko Budimir [b], Sven Lončarić [a]

[a] University of Zagreb, Faculty of Electrical Engineering and Computing, Zagreb, Croatia
[b] Institute for Nuclear Technologies (INETEC), Zagreb, Croatia

ARTICLE INFO

Article history:
Received 30 June 2021
Revised 24 November 2021
Accepted 4 December 2021
Available online 10 December 2021
Communicated by Zidong Wang

Keywords:
Image analysis
Convolutional neural networks
Non-destructive testing
Ultrasonic imaging
Defect detection

ABSTRACT

Non-destructive testing (NDT) is a set of techniques used for material inspection and detection of defects. Ultrasonic testing (UT) is one of the NDT techniques, commonly used to inspect components in the oil and gas industry, aerospace, and various types of power plants. Acquisition of the UT data is currently done automatically using robotic manipulators. This ensures the precision and uniformity of the acquired data. On the other hand, the analysis is still done manually by trained experts. Since the acquired UT data can be represented in the form of images, computer vision algorithms can be applied to analyze the content of images and localize defects. In this work, we propose a novel deep learning architecture designed specifically for defect detection from UT images. We propose a lightweight feature extractor that improves the precision and efficiency of the detector. We also modify the detection head to improve the detection of the objects with extreme aspect ratios which are common in UT images. We tested our approach on an in-house dataset with over 4000 images. The proposed architecture outperformed the previous state-of-the-art method by 1.7% (512 × 512 px input resolution) and 2.7% (384 × 384 px input resolution) while significantly decreasing the inference time.

1. Introduction

Non-destructive testing (NDT) is a popular approach for material evaluation and defect detection [1]. It is used for continuous inspection in numerous domains but most commonly in oil and gas industries, power and energy industries, aerospace, and construction. NDT includes a variety of techniques such as ultrasonic, eddy current, thermography, and x-radiography, to name a few. Each of the methods comes with its own advantages and disadvantages and they are sometimes also used jointly in order to increase the probability of finding a defect. None of the NDT techniques cause any damage to the inspected material so the tested component can normally be used after the inspection (if no problems were found) or sometimes even during the inspection. Ultrasonic testing (UT) is one of the most used NDT methods for detection, localization and measurement of flaws present in engineering materials under inspection [2]. UT is simple to perform, yields a precise location of the defect, and in general has a high signal-to-noise ratio [3]. Inspection is performed by the generation and detection of mechanical vibrations or waves within test objects [4]. There are several ways how this can be done. Pulse-echo (PE), time-of-flight-diffraction (TOFD), and phased array systems are the standard three implementations. A phased array system is a multi-channel ultrasonic system, which uses the principle of a time-delayed triggering of the transmitting transducer elements, combined with a time corrected receiving of detected signals [5]. Using the phased array it is possible to inspect the material from various angles at the same time, which is the main advantage compared to other types of UT probes. Inspecting the component using different angle values makes the process more reliable but it also produces huge amounts of data. Fig. 1 illustrates the principle behind phased array system inspection. Data from UT inspection can be displayed in different forms. As the probe is moved along the surface of the inspected material, at each position it transmits and receives ultrasound waves. The energy of the received ultrasound signal can be shown as a function of time in a representation called A-scan. Each A-scan can be converted into one image column so multiple A-scans can be stacked to form an image representation called B-scan. Since the ultrasound waves are often transmitted at some specific angle, A-scans can also be transferred onto the image at that angle. A view created this way is called

* Corresponding author.
E-mail addresses: duje.medak@fer.hr (D. Medak), luka.posilovic@fer.hr (L. Posilović), marko.subasic@fer.hr (M. Subašić), marko.budimir@inetec.hr (M. Budimir), sven.loncaric@fer.hr (S. Lončarić).

https://doi.org/10.1016/j.neucom.2021.12.008
0925-2312/

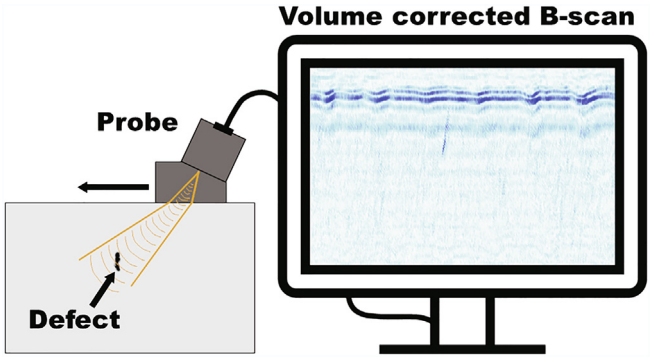


Fig. 1. Illustration of phased array system inspection. An example of volume corrected B-scan (VC-B-scan) is shown on the right side of the figure.

volume-corrected-B-scan (VC-B-scan) and it is the one used in this work.

Data acquired during the UT inspection still has to be analyzed manually by trained experts. This process is laborious and time-consuming. The number of ultrasonic inspections is increasing because most of the existing components require more inspections as time passes and the chance of defect occurrence increases. This also increases the need for optimizing the process of data analysis. Automated analysis can be used to improve the reliability when performing a manual inspection or to speed up the analysis by several orders of magnitude if used independently. The idea of automated analysis of UT data is not new, but the methods proposed so far are not reliable enough to be used in real-life situations. Some of the problems encountered when developing automated analysis of UT data include difficulty with the acquisition of large and diverse datasets, noise, and irregular signal appearances caused by odd defects' shapes and geometry of the inspected component. Recently an improvement in automated UT image analysis was made by employing deep learning approaches for classification and object detection. If an existing architecture is used for object detection, it is assumed that the shapes of the objects that need to be detected will be similar to the common objects found in PASCAL VOC [6] and COCO [7] datasets. Taking into consideration aspect ratios of the objects that need to be detected is very important and in some cases [8,9], proper design of architecture and training procedure leads to improved results. The goal of this work is to design an architecture that can precisely localize defects from B-scans obtained with a phased array probe. The usage of such probes is increasing in real-life inspections and a proper method for analysis of the collected data would be very useful. Depending on the inspected configuration and material, defects' signals can appear very elongated. This can make training difficult because popular anchor-based object detectors [10–12] have a limited number of anchors that are distanced from each other by a fixed value (stride). Having an extreme aspect ratio (>4) leads to a small overlap between the neighboring anchors thus reducing the coverage of an image. This decreases the number of sampled anchors used during the training which can have a negative impact on the detector's performance.

In this work, we propose a deep learning object detector to analyze VC-B-scans and localize all of the visible defects. We start from the state-of-the-art object detection architecture EfficientDet [12] and revise the building components of this model. We first replace the originally used EfficientNet [13] network with our custom feature extraction network. A new model is more precise and uses drastically fewer parameters leading to a faster prediction process. We then redesign the detection head in order to account for extreme aspect ratios that appear in UT images. We propose the usage of asymmetrical feature maps as inputs to the detection head in combination with lower template anchors stride. This increases the overlap between the template anchors and the ground truth labels and leads to a better model performance with a small computational overhead. The final object detector proposed in this work achieves a mean average precision of 91.3% which is 1.7% more than the previous state-of-the-art model EfficientDet-D0 [14]. Furthermore, the proposed model reduces the needed inference time by more than 30% and has 6 times fewer parameters compared to EfficientDet-D0.

1.1. Contributions

The main contributions of this work are the following:

- A novel feature extractor for the EfficientDet that improves the precision while using six-time fewer parameters.
- A method for detection of objects with extreme aspect ratios based on a modified detection head and dense placement of the anchors.
- A novel deep learning architecture created by joining aforementioned components into a new model that outperforms the previous state-of-the-art in defect detection in ultrasonic images.

1.2. Related work

Analyzing NDT data is a time-consuming process prone to human errors since it depends solely on the experience and the knowledge of the person performing the analysis. In order to assist the experts during the analysis, various methods for defect detection were proposed throughout the years. Developed methods can work with different types of NDT data such as the data acquired during a visual inspection [15,16], thermography inspection [17–19], radiography inspection [20,21], or ultrasonic inspection [22–27]. While the exact implementation depends on the used inspection technique and material, approaches for data analysis and ideas behind them are usually similar. Most of the recent methods rely on convolutional neural networks (CNNs) [15–25,27] since this type of architecture works well with one-dimensional and two-dimensional data such as sequences and images. It was shown that CNNs outperform classical approaches based on hand-crafted features in many general computer vision challenges like PASCAL [6], COCO [7], or ImageNet [28]. The authors of several works [15,3,16,22] tested this hypothesis for NDT data and came to the same conclusion that deep learning approaches outperform classical approaches.

Acquiring the data with non-destructive testing can be a costly process. The equipment required for inspection, as well as the examples of materials containing realistic flaws, are usually very expensive. Since only a fraction of the collected data represents defect signals, collecting a large set of useful images is difficult. This drawback can be solved in three ways: (I) Analysis of A-scans instead of B-scans (II) Application of traditional methods for image analysis that do not require a large dataset (III) Generating or simulating images that can be used to develop a modern deep learning model. The main problem with the A-scan analysis is the lack of context from the surrounding area which makes the decision-making process difficult. The most popular approach for defect detection from A-scans is using the wavelet transform to calculate features and then classifying extracted features using support vector machines (SVM) [29–31] or artificial neural networks (ANN) [32,33]. This way the available data is used solely for classifier training since the feature extraction is predefined. If the available dataset of B-scans is not big enough, some traditional approaches can be used but their performance and generalization are usually not as good as in deep learning approaches. In [34], the authors used the adaptive histogram equalization technique

followed by morphological operations to separate the defective zones from the non-defective zones in the ultrasonic TOFD images. Analyzing TOFD images was also the topic in [35,36] where the authors showed how the parabola matched filter and Hough transform can be used to locate parabolas in TOFD B-scans. In [37], Radon transform was used to detect defects from B-scans that were denoised using the wavelet transform. Having several hundred images already allows for deep learning methods to be employed. In that case, a CNN can be trained with the help of transfer learning [38] and data augmentation. This approach proved to be useful for defect detection from UT images [25,14] and X-ray images [19,20]. Another approach is to use simulated [23,39] or generated [21,24,40] data. While these types of images can be useful for model training, evaluation should be performed on a real dataset to ensure the credibility of the obtained results. In [41] the authors used a generated dataset of B-scans to train a VGG-like classification model. They tested the performance on a separate dataset of real B-scans and reported results almost as good as the one achieved by the human inspectors. The equipment used for acquisition in that work is quite similar to the one used for the collection of the dataset in our work but the tested specimen is different. In [42], the authors tested several deep learning classifiers. The dataset was acquired by a pulsed laser that transmits ultrasonic waves through the material while the contact transducer which is attached to the scanned object captures a series of snapshots of the propagating waves. Among the tested classifiers DenseNet [43] achieved the best result reaching an f1 score of 95.33%.

Defect detection from images can be done on various localization granularity levels. Some of the work [15,17,16,21,24,23] only determine if an image contains a defect or not. This is usually done by employing one of the popular image classification architectures such as VGG [44], Inception [45–47], ResNet [48,49], MobileNet [50–52], or by building a custom CNN. Other works [19,20,25,14] use approaches that determine a coarse location of the defect. This can be done by using object detection architectures which are usually divided into two families: One-stage detectors [53,54,11,12] and two-stage detectors [55–57]. Finally, a fine-grained localization (pixel-wise) can also be obtained as an output [17,18] if a model for semantic segmentation such as U-net [58] is used.

Having a coarse defect location is often good enough. In that case, using an object detection model instead of a semantic segmentation model is better since the inference time for object detectors is usually smaller. In this work, we use EfficientDet [12] architecture as a starting point. This state-of-the-art one-stage object detector was proven to work well with UT images [14]. We change the building blocks of the EfficientDet model by proposing a novel feature extraction network which we use instead of the standard EfficientNet [13] backbone. We also propose a modification of the model's detection head in order to improve the detection rate of objects with extreme aspect ratios. The description of the proposed components is given in Section 3.

2. Dataset

The architecture proposed in this work is developed for defect detection from ultrasonic images. The dataset was obtained by scanning six steel blocks with a phased array probe. Some of the images had a lot bigger width compared to their height. This can cause problems after padding and resizing images to input resolution so we cropped those types of images into multiple patches. Before the cropping is performed a desired width of the patches must be determined. In our case, the desired width was equal to the image height (all of the images that required cropping had a height of 375 px) since we wanted to get patches with an aspect ratio closest to one. We then divided an image into patches such that the obtained patches have the width as close as possible to the desired width. We also allowed the overlap of 20% between the neighboring patches. The final dataset contains more than 4000 VC-B-scans. The distribution of widths and heights of the images is shown in Fig. 2. The blocks contain 68 defects and each defect can be seen in multiple VC-B-scans (e.g. in various angles or scanning directions). All of the scans combined contain 6637 annotated defects. We do not distinguish between different types of defects so all of them are labeled with the same class. We divided the data into five folds where each fold contains unique defects. All of the appearances of a defect are placed into the same fold to ensure the credibility of the results during the cross-validation. More details about the dataset can be found in [14]. We used the same split so that we could compare results with the previous state-of-the-art approach.

3. Methodology

Deep learning object detectors are usually divided into two categories: one-stage detectors and two-stage detectors. Two-stage detectors used to be more precise but slower compared to the one-stage detectors. The accuracy gap between these two families of object detectors was decreased recently when new one-stage architectures were proposed [12,59]. In [14], it was shown that EfficientDet-D0 is the best choice among tested methods for defect detection from UT images. However, EfficientDet architecture was developed for general object detection on public datasets like [7], PASCAL VOC [6], or ImageNet [28]. Even though EfficientDet achieves good results when used for defect detection in ultrasonic images, we demonstrate that task-specific knowledge can be used to develop an even faster and more precise model.

3.1. Backbone design

General deep learning object detection architectures usually consist of three parts: feature extractor (backbone), feature network (detection neck), and detection head. The first part of the network is used to extract the features from the images. Feature extractors contain millions of parameters even if a simple network such as EfficientNet is used. Having a complex backbone ensures

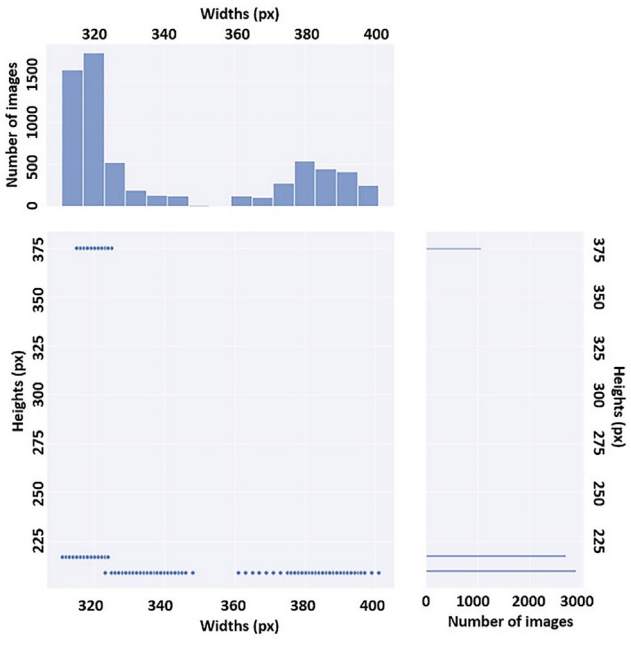


Fig. 2. Distribution of widths and heights of the images from the dataset.

that the extracted features are discriminative. This is very important when the task is to distinguish between dozens of complex objects which is a common requirement in popular object detection datasets. In this work, the goal is to detect only one class (defect). It was shown in [14] that simpler feature extractors can perform better than complex ones. This is why we decided to swap a standard EfficientNet architecture with a simpler novel network illustrated in Fig. 3. We propose an encoder-decoder type of network that looks similar to the U-net [58]. However, there are some key differences between the U-net and the architecture that we propose: (I) The proposed architecture does not use multiple blocks for each resolution level. (II) We do not increase the number of filters as the resolution is decreased. We use 32 filters in the first encoder-decoder part and 64 filters when creating feature maps used as input to the feature network. (III) Our blocks also contain batch normalization and dropout layers which help with the network regularization and improve results. (IV) Recently, to provide high performance of a deep network, several activation functions have been applied in different works [60–63]. However, they can lead to high computational costs and have been applied with different types of images rather than UT images. Instead of ReLu activation, our model uses the swish activation function (used in EfficientDet). The proposed network first downsamples the input features by performing a series of convolutions followed by the max-pooling operation. The decoding part of the network is similar to the one used in Hourglass networks [64]. The feature maps are first upsampled using the nearest-neighbor interpolation. We then perform addition with the feature map of the same resolution from the encoder and pass the resulting feature map through the activation function. We then perform 1x1 convolution followed by batch normalization and activation before upsampling the layer again. Once the original input resolution is reached the feature maps are downsampled again to create feature maps (P3-P7) that are used as an input to the feature network (bidirectional FPN used in EfficientDet [12]). Features P6 and P7 are not actually a part of the backbone. We calculated them using the same implementation used for their calculation in EfficientDet architecture.

If the EfficientNet backbone is replaced by the architecture proposed in this section, the total number of detector parameters is reduced from 3.88 million to 0.53 million. We showed in Section 4.2 that the proposed backbone increases the accuracy while simultaneously decreasing the inference time.

3.2. Detection head design

Many state-of-the-art methods use default boxes (anchors) as a rough starting shape to encapsulate objects and then they perform an extra step to fit the predicted boxes around the object more tightly. The shape of the anchors is determined from the hyperparameters. Since popular object detection datasets display everyday objects, there is no need for extremely shaped anchors (for example extreme aspect ratio or scale). This is why popular deep learning object detectors are designed to work only with standard anchor shapes or slightly modified ones. The problem with defect detection from ultrasonic images is the extreme aspect ratio of the objects. This can partially be solved by proper calculation of aspect ratio and scales hyperparameters as shown in [14]. In this work, we adopted the same aspect ratios and scales values used in that work. However, setting these hyperparameters does not solve another core problem that appears when using extreme aspect ratios which is the default placement of anchors. The default anchors use stride that is four times smaller than their size (for example feature map P5 uses a default size of anchor 128x128 and strides of 32x32). When common aspect ratios are used, this stride is sufficient to get proper coverage of the image meaning that the template anchors will overlap and no parts of the image will be left uncovered. However, if an extreme aspect ratio is used, a stride that is four times smaller is not sufficient.

We show in Fig. 4 how some of the used anchors with extreme aspect ratios appear once they are placed over the image. We plotted the placement for feature map P5 because it has fewer anchors compared to P3 and P4 so the image is concise and clear. It can be seen from the image that vertical gaps appear for these values of anchors which makes the detection harder. This problem can be solved by introducing more anchors with reduced horizontal spacing (stride) between them. This requires modification of the architecture. Originally used feature maps do not have sufficient resolution to increase the number of anchors so the feature maps of higher resolutions are needed. To accomplish this we shifted the input to the feature network (biFPN). This way the feature maps input to the detection head will also have a sufficient resolution. We use feature maps P2-P4 from the original network and the last two feature maps we calculated the same way that was used to calculate P6 and P7 in the original EfficientDet architecture. Since the defects from ultrasonic images are always elongated in the same direction (vertical), the reduction of stride is not needed in both directions. We inserted extra convolutional layers before the detection head to create asymmetrical feature maps. These convolutional layers downsample the height of the feature map so that the stride in vertical orientation could be left unchanged. Network which is modified to deal with detection of extreme aspect ratio objects is shown in Fig. 5. Proposed modifications can be used regardless of the chosen backbone. In Section 4.2 we showed that using the modified detection head improves the mean

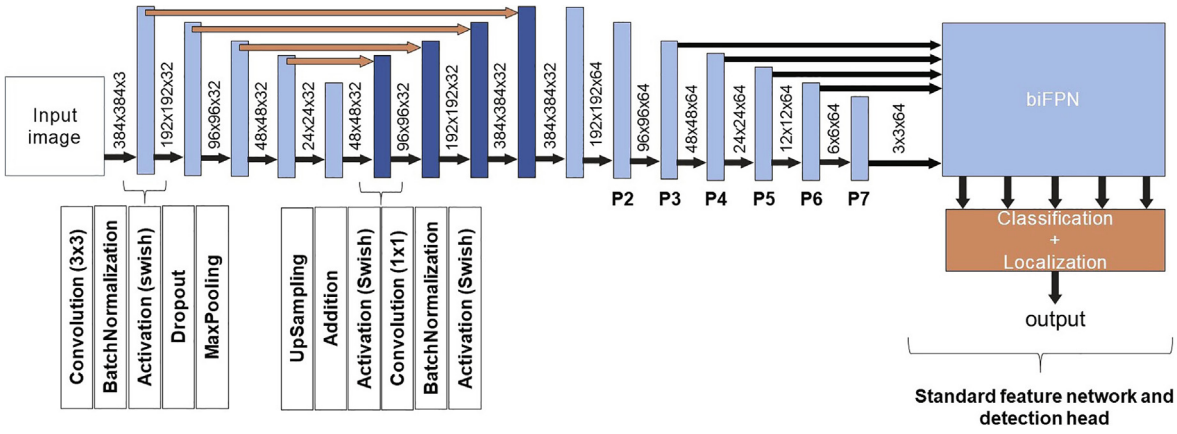


Fig. 3. The architecture of the proposed feature extraction network.

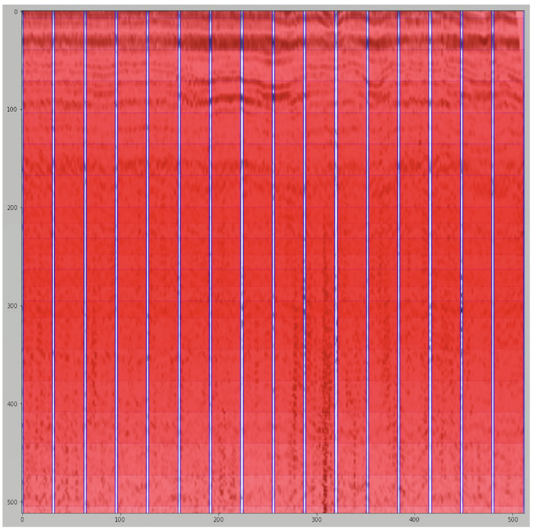


Fig. 4. Placement of anchors with extreme aspect ratios in standard detection head.

average precision on the defect detection task. Described modifications have a small computational overhead if used in combination with the custom feature extractor proposed in the previous section. If the modified detection head is used in combination with EfficientNet the inference time is actually decreased. This happens because the P5 feature map from EfficientNet does not need to be calculated so the number of parameters is reduced by almost three times.

4. Experimental setup and results

We evaluated the methods proposed in this work on our in-house dataset with over 4000 ultrasonic images containing defects. The dataset split and evaluation procedure are the same as in [14]. This allows us to compare obtained results with the top-performing method reported in that work which is EfficientDet-D0. Additionally, We compare the results of our DefectDet with the state-of-the-art family of object detectors YOLOv5 [65]. We run experiments that introduce proposed modules individually. First, we used the EfficientDet model but with the swapped backbone (using architecture introduced in Section 3.1 instead of EfficientNet). We then test the EfficientDet model but with the modified detection head that was proposed in Section 3.2. Finally, we join the two proposed modules into a new deep learning architecture that we named DefectDet. We train the proposed model using the focal loss [11]:

$$FL(p_t) = -(1 - p_t)^{\gamma} log(p_t) \quad (1)$$

We test the performance of individual modules and their combination with two input image resolutions: 512x512 pixels and 384x384 pixels. The training details are given in the next section.

Table 1
Impact of design choices on custom model with input size 384x384 pixels. The first row refer to EfficientDet-D0 from [14].

Custom backbone	Custom detection head	mAP	Inference time (ms)
		0.881	57.0
✓		0.894	39.1
	✓	0.891	45.6
✓	✓	0.908	40.3

Table 2
Impact of design choices on custom model with input size 512x512 pixels. The first row refer to EfficientDet-D0 from [14].

Custom backbone	Custom detection head	mAP	Inference time (ms)
		0.896	62.8
✓		0.900	39.5
	✓	0.902	49.4
✓	✓	0.913	41.7

4.1. Experimental setup

All of the models evaluated in this work were first pretrained on the COCO [7] dataset. As standard practice when using pretrained weights as a starting point, the input RGB images were normalized by subtracting the mean values (0.485, 0.456, 0.406) and dividing them with the standard deviations (0.229, 0.224, 0.225). There was no need for additional intensity normalization which is sometimes done when dealing with unnatural images such as ultrasonic images or magnetic resonance images [66,67]. RetinaNet, EfficientDet, and DefectDet were then trained using the ADAM opti-

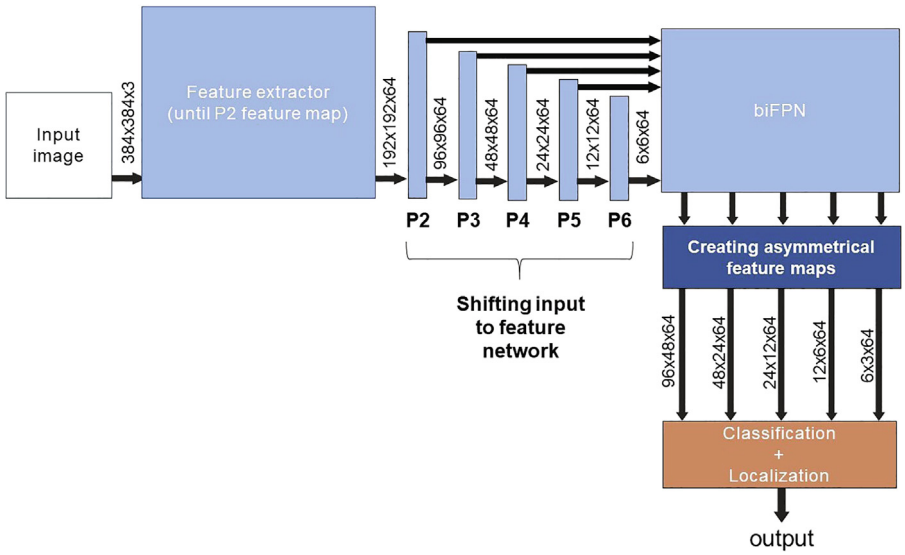


Fig. 5. Feature network and detection head that are customized to handle extreme aspect ratios.

Table 3
Mean average precision (mAP) and inference time for various architectures. Results for EfficientDet models and RetinaNet were taken from [14]. All of the models in the table were tested with input image of 512x512x3 except RetinaNet which achieves better results when the images are only padded.

Model	Fold1	Fold2	Fold3	Fold4	Fold5	Average	Inference time (ms)
EfficientDet-D0	0.937	0.829	0.879	0.943	0.893	0.896	67.2
EfficientDet-D1	0.927	0.793	0.869	0.917	**0.901**	0.881	75.2
EfficientDet-D2	0.936	0.780	0.826	0.920	0.895	0.871	76.4
RetinaNet	0.872	0.821	0.830	0.901	0.850	0.855	25.6
YOLOv5-s	0.926	0.853	0.827	0.946	0.830	0.876	**12.3**
YOLOv5-m	0.924	0.813	0.840	0.947	0.875	0.880	16.0
YOLOv5-l	0.922	0.861	0.869	0.944	0.852	0.890	19.5
YOLOv5-x	0.925	0.839	0.838	0.951	0.823	0.875	22.8
DefectDet	**0.942**	**0.869**	**0.903**	**0.956**	0.894	**0.913**	35.8

mizer with an initial learning rate of 10^{-3}. We left out 15% of the training subset for validation which was used to reduce the learning rate on a plateau and early stopping of the training. Smaller models (using 384x384 input) were trained with batch size 8 and 500 steps per epoch while the bigger models were trained with batch size 4 and 1000 steps per epoch. Batch size 16 was used for YOLOv5 models and the optimization was done using the SGD since it achieved better results compared to the ADAM optimizer. The rest of the hyperparameters for YOLOv5 were set to default values proposed by its creators. RetinaNet, EfficientDet, and DefectDet were implemented in the Keras library (version 2.2.5) using the Tensorflow backend (version 1.15.0). PyTorch version 1.9.0 was used when testing YOLOv5 models. Inference times from Table 1 and Table 2 were measured on a machine with Titan Xp GPU and CUDA 11.0. Inference times from Table 3 were measured on the same machine but the CUDA 11.2 version. We used mean average precision (mAP) averaged across 5 folds to evaluate the performance of the models. The results are shown and discussed in the following section.

4.2. Results and discussion

The results of the experiments are shown in Tables 1 and 2. The first row of the table corresponds to the EfficientDet-D0 that is the current state-of-the-art in the defect detection task. Swapping the EfficientNet backbone with the one proposed in this work improves the mean average precision (mAP) while simultaneously decreasing the inference time. The mAP is especially increased for the smaller model. Since the images from our datasets are all smaller than 400x400 pixels the difference between the performances of smaller and bigger models should not be big.

For the original EfficientDet architecture the difference between the models of lower (384×384) and higher (512×512) resolution was 1.5%. If the backbone proposed in this work is used this difference is reduced three times. This indicates that the proposed backbone was designed well and that more information is preserved when analyzing the images in their natural resolution. The third row shows the benefits of the modified detection head with asymmetrical feature map inputs and decreased stride. As explained in

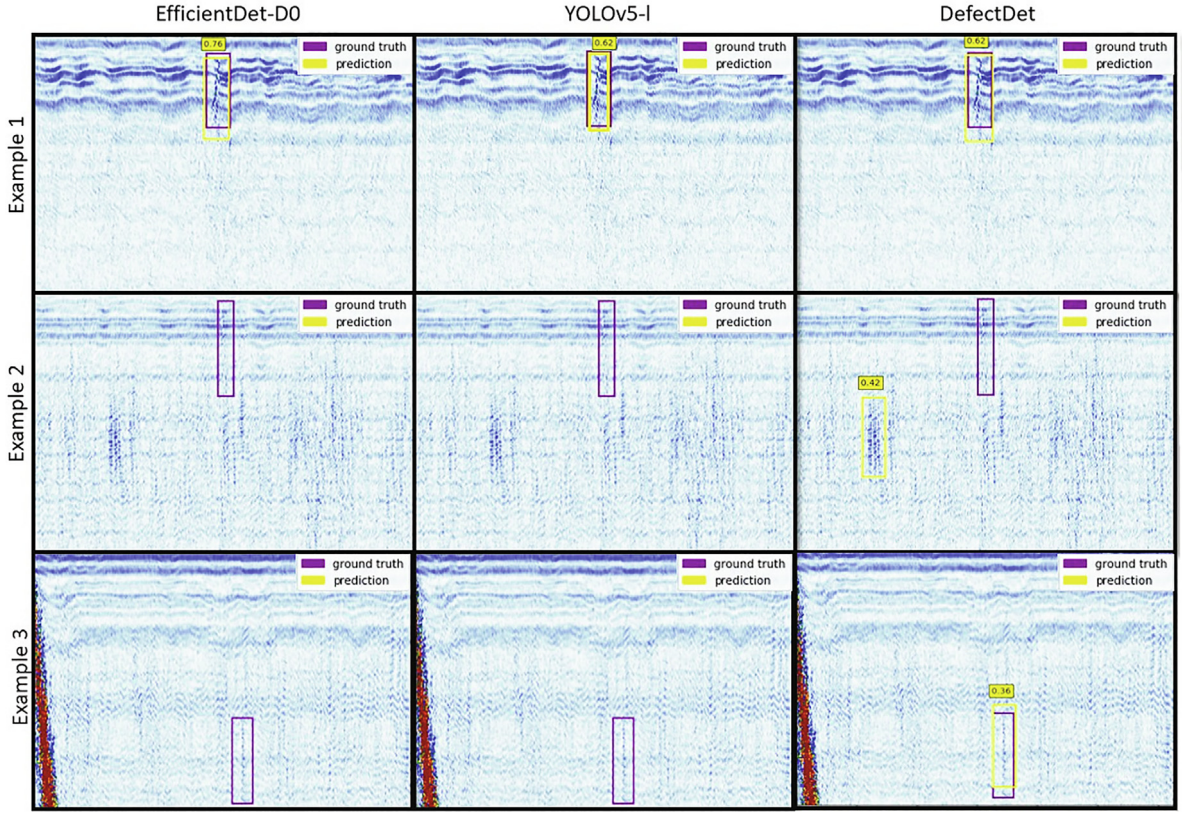


Fig. 6. A few examples of detection on the test images. The threshold for all of the models was 0.3.

Section 3.2, using the custom detection head in combination with EfficientNet actually decreases the inference time since the P5 feature map does not need to be calculated. Finally, the last rows show the performance of our DefectDet which was obtained by joining the custom backbone and custom detection head. If the custom feature extractor is used, a replacement of the standard detection head with the one proposed in this work will lead to a very small increase of inference time. However, the proposed architecture is still more than 30% faster compared to the baseline model EfficientDet-D0. At the same time, the mean average precision increases by 2.7% for the smaller model and 1.7% for the bigger model.

A comparison of the DefectDet with current state-of-the-art models is given in Table 3. A state-of-the-art model YOLOv5 achieves similar results as the previously tested EfficientDet. The proposed DefectDet architecture outperforms all the other tested models for each fold except the fourth fold. Even the DefectDet with input resolution 384×384 surpasses all of the other architectures with greater input resolution. The quickest model among the tested ones was YOLOv5-small but the mean average precision of that model is 3.7% lower than our DefectDet. We also tested the YOLOv5 family of models with a smaller input resolution ($384 \times 384 \times 3$) and all of the tested models achieved less than 87.5% of mAP which is significantly lower compared to the DefectDet with the same input resolution (90.8%). Some prediction examples can be seen in Fig. 6. Shown examples were randomly picked from the second fold test set which is the subset for which the models achieved the lowest mAP on average. None of the tested models were trained using the picked examples. All of the models successfully detected the defect on the first example image. The second example contains a signal that is barely visible and can not be detected without looking at surrounding B-scans so it is not surprising that all of the models failed to detect it. When tested on the third example, EfficientDet and YOLOv5 did not manage to detect a defect while DefectDet managed. Looking at the example images one can notice how hard it is to detect a defect in some of the images, especially when the image is noisier or when it contains geometry signals.

5. Conclusion

In this paper, we propose a novel architecture for detecting defects from ultrasonic images. We designed a simple feature extraction network that enables quicker and more precise detection of defects compared to the previously used models. The proposed feature extractor also reduces the difference in performance between models with different input image resolutions. Furthermore, we proposed a solution to improve the detection of objects with extreme aspect ratios by altering the detection head of the model. With these changes introduced, our defect detection framework outperformed the previous state-of-the-art baseline, and at the same time, required fewer parameters, thus reducing memory usage and inference time. Compared to the state-of-the-art EfficientDet-D0 model, our architecture improves mean average precision by 1.7% for the bigger model, and 2.7% for the smaller model while simultaneously decreasing the inference time by more than 30%. Even though the developed architecture was designed for a specific application, we believe that the proposed ideas can be generalized to other domains with similar problems as defect detection.

CRediT authorship contribution statement

Duje Medak: Conceptualization, Methodology, Software, Validation, Data curation, Writing – original draft. **Luka Posilović:** Software, Data curation, Writing – review & editing. **Marko Subašić:** Conceptualization, Resources, Writing – review & editing, Supervision. **Marko Budimir:** Resources, Data curation, Writing – review & editing, Funding acquisition. **Sven Lončarić:** Resources, Writing – review & editing, Supervision, Funding acquisition.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research was co-funded by the European Union through the European Regional Development Fund, under the grant KK.01.2.1.01.0151 (Smart UTX). We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

References

[1] L. Cartz, Nondestructive testing: radiography, ultrasonics, liquid penetrant, magnetic particle, eddy current, ASM International, 1995. URL: https://books.google.hr/books?id=0spRAAAAMAAJ..
[2] J. Veiga, A.A. de Carvalho, I. Silva, J.M.A. Rebello, The use of artificial neural network in the classification of pulse-echo and tofd ultra-sonic signals, Journal of The Brazilian Society of Mechanical Sciences and Engineering - J BRAZ SOC MECH SCI ENG 27. doi:10.1590/S1678-58782005000400007..
[3] J. Ye, S. Ito, N. Toyama, Computerized ultrasonic imaging inspection: From shallow to deep learning, Sensors 18 (11) (2018) 3820, https://doi.org/10.3390/s18113820.
[4] D. Forsyth, 5 - nondestructive testing of corrosion in the aerospace industry, in: S. Benavides (Ed.), Corrosion Control in the Aerospace Industry, Woodhead Publishing Series in Metals and Surface Engineering, Woodhead Publishing, 2009, pp. 111–130. doi:10.1533/9781845695538.2.111..
[5] A. Bulavinov, D. Joneit, M. Kroening, L. Bernus, M. Dalichow, K. Reddy, Sampling phased array - a new technique for signal processing and ultrasonic imaging, Fraunhofer IZFP-D 48. doi:10.1117/12.717891..
[6] M. Everingham, L. van Gool, C. Williams, J. Winn, A. Zisserman, The PASCAL Object Recognition Database Collection. URL: http://host.robots.ox.ac.uk/pascal/VOC/, [Online; accessed 1-May-2020] (2012)..
[7] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C.L. Zitnick, Microsoft coco: Common objects in context, in: D. Fleet, T. Pajdla, B. Schiele, T. Tuytelaars (Eds.), Computer Vision – ECCV 2014, Springer International Publishing, Cham, 2014, pp. 740–755.
[8] D. Kim, S. Kim, S. Jeong, J.-W. Ham, S. Son, J. Moon, K.-Y. Oh, Rotational multipyramid network with bounding-box transformation for object detection, International Journal of Intelligent Systems 36 (9) (2021) 5307–5338. doi:10.1002/int.22513..
[9] J.-B. Hou, X. Zhu, X.-C. Yin, Self-adaptive aspect ratio anchor for oriented object detection in remote sensing images, Remote Sens. 13 (7). doi:10.3390/rs13071318..
[10] J. Redmon, A. Farhadi, Yolo9000: Better, faster, stronger, CoRR abs/1612.08242. url:http://arxiv.org/abs/1612.08242..
[11] T. Lin, P. Goyal, R.B. Girshick, K. He, P. Dollár, Focal loss for dense object detection, CoRR abs/1708.02002. arXiv:1708.02002. url:http://arxiv.org/abs/1708.02002.
[12] M. Tan, R. Pang, Q.V. Le, Efficientdet: Scalable and efficient object detection, ArXiv abs/1911.09070..
[13] M. Tan, Q.V. Le, Efficientnet: Rethinking model scaling for convolutional neural networks, CoRR abs/1905.11946. arXiv:1905.11946. url:http://arxiv.org/abs/1905.11946.
[14] D. Medak, L. Posilović, M. Subašić, M. Budimir, S. Lončarić, Automated defect detection from ultrasonic images using deep learning, IEEE Trans. Ultrason. Ferroelectr. Freq. Control (2021), https://doi.org/10.1109/TUFFC.2021.3081750, 1–1.
[15] J. Masci, U. Meier, D. Ciresan, J. Schmidhuber, G. Fricout, Steel defect classification with max-pooling convolutional neural networks, in: The 2012 International Joint Conference on Neural Networks (IJCNN), 2012, pp. 1–6, https://doi.org/10.1109/IJCNN.2012.6252468.
[16] Y. Yu, H. Cao, X. Yan, T. Wang, S.S. Ge, Defect identification of wind turbine blades based on defect semantic features with transfer feature extractor, Neurocomputing 376 (2020) 1–9, https://doi.org/10.1016/j.neucom.2019.09.071.
[17] Q. Luo, B. Gao, W. Woo, Y. Yang, Temporal and spatial deep learning network for infrared thermal defect detection, NDT & E Int. 108 (2019), https://doi.org/10.1016/j.ndteint.2019.102164 102164.

[18] L. Ruan, B. Gao, S. Wu, W.L. Woo, Deftectnet: Joint loss structured deep adversarial network for thermography defect detecting system, Neurocomputing 417 (2020) 441–457, https://doi.org/10.1016/j.neucom.2020.07.093.
[19] H.-T. Bang, S. Park, H. Jeon, Defect identification in composite materials via thermography and deep learning techniques, Compos. Struct. 246 (2020), https://doi.org/10.1016/j.compstruct.2020.112405 112405.
[20] W. Du, H. Shen, J. Fu, G. Zhang, Q. He, Approaches for improvement of the x-ray image defect detection of automobile casting aluminum parts based on deep learning, NDT & E Int. 107 (2019), https://doi.org/10.1016/j.ndteint.2019.102144 102144.
[21] X. Le, J. Mei, H. Zhang, B. Zhou, J. Xi, A learning-based approach for surface defect detection using small image datasets, Neurocomputing 408 (2020) 112–120, https://doi.org/10.1016/j.neucom.2019.09.107.
[22] M. Meng, Y.J. Chua, E. Wouterson, C.P.K. Ong, Ultrasonic signal classification and imaging system for composite materials via deep convolutional neural networks, Neurocomputing 257 (2017) 128–135, machine Learning and Signal Processing for Big Multimedia Analysis. doi:10.1016/j.neucom.2016.11.066..
[23] K. Virupakshappa, E. Oruklu, Multi-class classification of defect types in ultrasonic ndt signals with convolutional neural networks, in, IEEE International Ultrasonics Symposium (IUS) 2019 (2019) 1647–1650, https://doi.org/10.1109/ULTSYM.2019.8926027.
[24] I. Virkkunen, T. Koskinen, O. Jessen-Juhler, J. Rinta-aho, Augmented ultrasonic data for machine learning, J. Nondestr. Eval. 40 (1) (2021) 1–11.
[25] L. Posilović, D. Medak, M. Subašić, T. Petković, M. Budimir, S. Lončarić, Flaw detection from ultrasonic images using yolo and ssd, in: 2019 11th International Symposium on Image and Signal Processing and Analysis (ISPA), IEEE, 2019, pp. 163–168..
[26] N. Munir, H.-J. Kim, S.-J. Song, S.-S. Kang, Investigation of deep neural network with drop out for ultrasonic flaw classification in weldments, J. Mech. Sci. Technol. 32 (7) (2018) 3073–3080, https://doi.org/10.1007/s12206-018-0610-1.
[27] N. Munir, H.-J. Kim, J. Park, S.-J. Song, S.-S. Kang, Convolutional neural network for ultrasonic weldment flaw classification in noisy conditions, Ultrasonics 94 (2019) 74–81, https://doi.org/10.1016/j.ultras.2018.12.001.
[28] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A.C. Berg, L. Fei-Fei, ImageNet Large Scale Visual Recognition Challenge, Int. J. Comput. Vis. 115 (3) (2015) 211–252, https://doi.org/10.1007/s11263-015-0816-y.
[29] V. Matz, M. Kreidl, R. Smid, Classification of ultrasonic signals, International Journal of Materials 27 (2006) 145–. doi:10.1504/IJMPT.2006.011267..
[30] A. Al-Ataby, W. Al-Nuaimy, C. Brett, O. Zahran, Automatic detection and classification of weld flaws in tofd data using wavelet transform and support vector machines, Insight - Non-Destructive Testing and Condition Monitoring 52 (2010) 597–602, https://doi.org/10.1784/insi.2010.52.11.597.
[31] Y. Chen, H.-W. Ma, G.-M. Zhang, A support vector machine approach for classification of welding defects from ultrasonic signals, Nondestruct. Test. Eval. 29(3) (2014) 243–254. doi:10.1080/10589759.2014.914210..
[32] F. Bettayeb, T. Rachedi, H. Benbartaoui, An improved automated ultrasonic nde system by wavelet and neuron networks, Ultrasonics 42(1) (2004) 853–858, proceedings of Ultrasonics International 2003. doi:10.1016/j.ultras.2004.01.064..
[33] S. Sambath, P. Nagaraj, N. Selvakumar, Automatic defect classification in ultrasonic ndt using artificial intelligence, J. Nondestr. Eval. 30 (1) (2011) 20–28, https://doi.org/10.1007/s10921-010-0086-0.
[34] T. Merazi-Meksen, M. Boudraa, B. Boudraa, Ultrasonic image enhancement to internal defect detection during material inspection, in: MATEC Web of Conferences, vol. 208, EDP Sciences, 2018, p. 01005..
[35] P. Petcher, S. Dixon, Parabola detection using matched filtering for ultrasound b-scans, Ultrasonics 52 (1) (2012) 138–144.
[36] P. Bolland, L. Lew Yan Voon, B. Gremillet, L. Pillet, A. Diou, P. Gorria, The application of hough transform on ultrasonic images for the detection and characterization of defects in non-destructive inspection, in: Proceedings of Third International Conference on Signal Processing (ICSP'96), Vol. 1, 1996, pp. 393–396 vol 1. doi:10.1109/ICSIGP.1996.567285..
[37] H. Cygan, L. Girardi, P. Aknin, P. Simard, B-scan ultrasonic image analysis for internal rail defect detection, World Congress on Railway Research, 2003.
[38] A.S. Razavian, H. Azizpour, J. Sullivan, S. Carlsson, Cnn features off-the-shelf: An astounding baseline for recognition, IEEE Conference on Computer Vision and Pattern Recognition Workshops 2014 (2014) 512–519, https://doi.org/10.1109/CVPRW.2014.131.
[39] R.J. Pyle, R.L.T. Bevan, R.R. Hughes, R.K. Rachev, A.A.S. Ali, P.D. Wilcox, Deep learning for ultrasonic crack characterization in nde, IEEE Trans. Ultrason. Ferroelectrics Freq. Control (2020) 1–1 doi:10.1109/TUFFC.2020.3045847..
[40] L. Posilović, D. Medak, M. Subasic, M. Budimir, S. Loncaric, Generative adversarial network with object detector discriminator for enhanced defect detection on ultrasonic b-scans (2021). arXiv:2106.04281..
[41] O. Siljama, T. Koskinen, O. Jessen-juhler, I. Virkkunen, Automated flaw detection in multi-channel phased array ultrasonic data using machine learning, J. Nondestruct. Eval. 40(3), funding Information: Welds were contributed by Suisto Engineering. UT data scanning was contributed by DEKRA. Data augmentation was contributed by Trueflaw. Their support is gratefully acknowledged. Publisher Copyright: 2021, The Author(s). doi:10.1007/s10921-021-00796-4..
[42] J. Ye, N. Toyama, Benchmarking deep learning models for automatic ultrasonic imaging inspection, IEEE Access 9 (2021) 36986–36994, https://doi.org/10.1109/ACCESS.2021.3062860.
[43] G. Huang, Z. Liu, K.Q. Weinberger, Densely connected convolutional networks, CoRR abs/1608.06993. arXiv:1608.06993. url:http://arxiv.org/abs/1608.06993.
[44] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, CoRR abs/1409.1556. url:https://arxiv.org/abs/1409.1556.
[45] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, CoRR abs/1512.00567. url:http://arxiv.org/abs/1512.00567.
[46] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in: CVPR, IEEE Computer Society, 2016, pp. 2818–2826. doi:10.1109/CVPR.2016.308..
[47] C. Szegedy, S. Ioffe, V. Vanhoucke, Inception-v4, inception-resnet and the impact of residual connections on learning, CoRR abs/1602.07261. arXiv:1602.07261. url:http://arxiv.org/abs/1602.07261.
[48] K. He, X.Z. 0006, S. Ren, J.S. 0001, Deep residual learning for image recognition, CoRR abs/1512.03385. url:http://arxiv.org/abs/1512.03385.
[49] S. Xie, R.B. Girshick, P. Dollár, Z. Tu, K. He, Aggregated residual transformations for deep neural networks, CoRR abs/1611.05431. arXiv:1611.05431. url:http://arxiv.org/abs/1611.05431.
[50] A.G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, Mobilenets: Efficient convolutional neural networks for mobile vision applications, CoRR abs/1704.04861. arXiv:1704.04861. url:http://arxiv.org/abs/1704.04861.
[51] M. Sandler, A.G. Howard, M. Zhu, A. Zhmoginov, L. Chen, Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation, CoRR abs/1801.04381. arXiv:1801.04381. url:http://arxiv.org/abs/1801.04381.
[52] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, Q.V. Le, H. Adam, Searching for mobilenetv3, in: The IEEE International Conference on Computer Vision (ICCV), 2019..
[53] J. Redmon, S.K. Divvala, R.B.Girshick, A. Farhadi, You only look once: Unified, real-time object detection, in: CVPR, IEEE Computer Society, 2016, pp. 779–788. doi:10.1109/CVPR.2016.91..
[54] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A. Berg, Ssd: Single shot multibox detector, Vol. 9905, 2016, pp. 21–37. doi:10.1007/978-3-319-46448-0_2..
[55] R. Girshick, Fast r-cnn, in: Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), ICCV '15, IEEE Computer Society, USA, 2015, p. 1440–1448. doi:10.1109/ICCV.2015.169..
[56] S. Ren, K. He, R.B. Girshick, J.S. 0001, Faster r-cnn: Towards real-time object detection with region proposal networks, IEEE Trans. Pattern Anal. Mach. Intell 39 (6) (2017) 1137–1149. doi:10.1109/TPAMI.2016.2577031..
[57] T.-Y. Lin, P. DollÃ!'r, R.B. Girshick, K. He, B. Hariharan, S.J. Belongie, Feature pyramid networks for object detection, CoRR abs/1612.03144. url:http://arxiv.org/abs/1612.03144.
[58] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: N. Navab, J. Hornegger, W.M. Wells, A.F. Frangi (Eds.), Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, 2015, pp. 234–241. doi:10.1007/978-3-319-24574-4_28..
[59] A. Bochkovskiy, C.-Y. Wang, H.-Y.M. Liao, Yolov4: Optimal speed and accuracy of object detection, ArXiv abs/2004.10934..
[60] E. Goceri, Analysis of deep networks with residual blocks and different activation functions: Classification of skin diseases, in: 2019 Ninth International Conference on Image Processing Theory, Tools and Applications (IPTA), 2019, pp. 1–6, https://doi.org/10.1109/IPTA.2019.8936083.
[61] L. Nanni, A. Lumini, S. Ghidoni, G. Maguolo, Stochastic selection of activation layers for convolutional neural networks, Vol. 20, 2020. doi:10.3390/s20061626..
[62] E. Goceri, Diagnosis of skin diseases in the era of deep learning and mobile technology, Comput. Biol. Med. 134 (2021), https://doi.org/10.1016/j.compbiomed.2021.104458 104458.
[63] E. Goceri, Deep learning based classification of facial dermatological disorders, Comput. Biol. Med. 128 (2021), https://doi.org/10.1016/j.compbiomed.2020.104118 104118.
[64] A. Newell, K. Yang, J. Deng, Stacked hourglass networks for human pose estimation, in: Computer Vision – ECCV 2016, Springer International Publishing, Cham, 2016, pp. 483–499..
[65] G. Jocher, A. Stoken, J. Borovec, NanoCode012, A. Chaurasia, TaoXie, L. Changyu, A. V, Laughing, tkianai, yxNONG, A. Hogan, lorenzomammana, AlexWang1900, J. Hajek, L. Diaconu, Marc, Y. Kwon, oleg, wanghaoyang0106, Y. Defretin, A. Lohia, ml5ah, B. Milanko, B. Fineran, D. Khromov, D. Yiwei, Doug, Durgesh, F. Ingham, ultralytics/yolov5: v5.0 - YOLOv5-P6 1280 models, AWS, Supervise.ly and YouTube integrations (Apr. 2021). doi:10.5281/zenodo.4679653..
[66] E. Goceri, Intensity normalization in brain mr images using spatially varying distribution matching, in: 11th Int. Conf. on computer graphics, visualization, computer vision and image processing (CGVCVIP 2017), 2017, pp. 300–4..
[67] E. Goceri, Fully automated and adaptive intensity normalization using statistical features for brain mr images, Celal Bayar Univ. J. Sci. 14 (1) (2018) 125–134.

Duje Medak received his M.Sc. from the University of Zagreb, Faculty of Electrical Engineering and Computing in 2019. He is currently pursuing a Ph.D. degree in the same faculty while working as a researcher in the Image Processing Group in the Department of Electronic Systems and Information Processing. His research interests include image processing, image analysis, machine learning, and deep learning. His current research interest includes deep learning object detection methods and their application in the non-destructive testing (NDT) domain.

Luka Posilović received his M.Sc. from the University of Zagreb, Faculty of Electrical Engineering and Computing in 2019. He is currently working as a young researcher in an Image Processing Group in the Department of Electronic Systems and Information Processing and working on his Ph.D. at the same University. His research interests include visual quality control, deep learning object detection, and synthetic image generation.

Marko Subašić is an associate professor at the Department for Electronic Systems and Information Processing, Faculty of Electrical Engineering and Computing, University of Zagreb, and has been working there since 1999. He received his Ph.D. degree from the Faculty of Electrical Engineering and Computing, University of Zagreb, in 2007. His field of research is image processing and analysis and neural networks with a particular interest in image segmentation, detection techniques, and deep learning

Marko Budimir received his M.Sc. of physics at University of Zagreb, Faculty of Science in 2000., and his Ph.D. at Ecole Polytechnique Federale de Lausanne in Switzerland om 2006. He worked at EPFL from 2006. till 2008. From 2008. he is working at the Institute of Nuclear Technology (INETEC). He coordinated many key projects at INETEC and although he is a key person in a company of industry sector he is still working close to the field of science.

Sven Lončarić is a professor of electrical engineering and computer science at the Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia. As a Fulbright scholar, he received a Ph.D. degree in electrical engineering from the University of Cincinnati, OH in 1994. From 2001–2003, he was an assistant professor at the New Jersey Institute of Technology, USA. His areas of research interest are image processing and computer vision. He was the principal investigator on a number of R&D projects. Prof. Lončarić co-authored more than 250 publications in scientific journals and conferences. He is the director of the Center for Computer Vision at the University of Zagreb and the head of the Image Processing Group. He is a co-director of the Center of Excellence in Data Science and Cooperative Systems. Prof. Lončarić was the Chair of the IEEE Croatia Section. He is a senior member of IEEE and a member of the Croatian Academy of Technical Sciences. Prof. Lončarić received several awards for his scientific and professional work.

Publication 3

L. Posilović, **D. Medak**, M. Subašić, M. Budimir, S. Lončarić, ”Generative adversarial network with object detector discriminator for enhanced defect detection on ultrasonic b-scans”, *Neurocomputing*, vol. 459, Oct. 2021, pp. 361-369.

Neurocomputing 459 (2021) 361–369

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Generative adversarial network with object detector discriminator for enhanced defect detection on ultrasonic B-scans

Luka Posilović [a,*], Duje Medak [a], Marko Subašić [a], Marko Budimir [b], Sven Lončarić [a]

[a] University of Zagreb, Faculty of Electrical Engineering and Computing, Zagreb, Croatia
[b] Institute for Nuclear Technologies (INETEC), Zagreb, Croatia

ARTICLE INFO

Article history:
Received 19 November 2020
Revised 29 April 2021
Accepted 28 June 2021
Available online 2 July 2021
Communicated by Zidong Wang

Keywords:
Non-destructive testing
Ultrasonic B-scan
Automated defect detection
Image generation
Generative adversarial networks

ABSTRACT

Non-destructive testing is a set of techniques for defect detection in materials. While the set of imaging techniques is manifold, ultrasonic imaging is the one used the most. The analysis is mainly performed by human inspectors manually analyzing the acquired images. A low number of defects in real ultrasonic inspections and legal issues concerning data from such inspections make it difficult to obtain proper results from automatic ultrasonic image (B-scan) analysis. The goal of presented research is to obtain an improvement of the detection results by expanding the training data set with realistic synthetic samples. In this paper, we present a novel deep learning Generative Adversarial Network model for generating realistic ultrasonic B-scans with defects in distinct locations. Furthermore, we show that generated B-scans can be used for synthetic data augmentation, and can improve the performances of deep convolutional neural object detection networks. Our novel method was developed on a dataset with almost 4000 images and more than 6000 annotated defects. When trained only on real data, detector can achieve an average precision of 70%. By training only on generated data the results increased to 72%, and by mixing generated and real data we achieve almost 76% average precision. We believe that synthetic data generation can generalize to other tasks with limited data. It could also be used for training human personnel.

1. Introduction

Non-destructive testing (NDT) is widely used in science and industry to evaluate properties of materials, components, or systems without causing damage [1]. Many different methods are available such as visual examination, ultrasonic, eddy current, to name a few. Among them, ultrasonic testing (UT) stands out due to its versatility. Some of the advantages of UT include high sensitivity for most of the materials [2], high signal to noise ratio [3] and the ability to precisely determine the location and the type of the defect [2]. Ultrasonic data can be represented in several different formats suitable for analysis including A, B, or C-scans [4]. An A-scan shows signal's amplitude as a function of time, B-scan displays a cross-sectional view of the inspected material, and a C-scan provides a top view of its projected features [5]. During analysis, inspectors simultaneously use multiple data representations in order to make a decision and evaluate the data.

Automated analysis has long been used in many NDT systems. However, so far it has been limited to classical decision-making algorithms such as amplitude thresholding [6]. Complex data such as the one from ultrasonic inspection makes it hard to develop an automated analysis. All ultrasonic analysis is, to the best of our knowledge, done manually by a trained human inspector. It makes ultrasonic analysis highly reliant on the inspector's experience. The automated analysis could make the process much faster and more reliable. There have been some attempts in developing an automated UT analysis [5–9], but very few of them involve using deep learning and modern deep convolutional neural networks (CNNs) on B-scans. The prerequisite for using deep learning is a large, annotated dataset. Due to a low number of flaws in real ultrasonic inspections and legal issues considering data from such inspections available data is limited. Data is the biggest drawback in the development of proper automated/assisted ultrasonic analysis. This challenge can also be found in many medical image analysis tasks [10] where, due to the rarity of some pathology and patient privacy issues, data availability is very modest. Furthermore, unlike medical datasets, there are no publicly available UT datasets.

Researchers attempt to overcome this problem by using transfer learning [11] in combination with freezing the backend CNN layers [12] which is shown to enhance the accuracy of models.

* Corresponding author.
E-mail addresses: luka.posilovic@fer.hr (L. Posilović), duje.medak@fer.hr (D. Medak), marko.subasic@fer.hr (M. Subašić), marko.budimir@inetec.hr (M. Budimir), sven.loncaric@fer.hr (S. Lončarić).

https://doi.org/10.1016/j.neucom.2021.06.094
0925-2312/

Using data augmentation is also the standard procedure for network training. However, data augmentation methods are limited and only slightly change some aspects of existing images (e.g. brightness modulation). Very limited additional information can be gained by such modifications. Synthetic data generation of high-quality images is a new type of state-of-the-art data augmentation [13]. Generative models such as generative adversarial networks (GANs) offer more variability and enrich the dataset to further improve the training process.

In this work, we present a novel GAN architecture for generating high-quality and realistic UT B-scans. Afterwards, we demonstrate that generated images can be used to train an object detection neural network to detect defects in real images. We show that images generated using our method improve the detector's average precision by more points than previous state-of-the-art augmentation techniques.

1.1. Contributions

The main contributions of this work are the following:

- a novel GAN architecture for generating high-quality ultrasonic images with objects at precise locations,
- experimental demonstration that expanding the ultrasonic dataset with generated synthetic data increases the performance of the defect detector,
- to the best of our knowledge, this is the first time a GAN is trained on ultrasonic NDT images.

1.2. Related work

Data availability is a major problem when using deep learning for defect detection. B-scans are the ideal data representation for accurately detecting defects and further estimating their depth and size. However, most authors focus on developing methods for A-scan analysis because it is easier to gather enough data. Developed algorithms for defect detection can be divided into three groups related to data representation being used; A-scans [7,8,14–23], B-scans [5,6,24,25] and C-scans [26,27]. The A-scan analysis is the most researched group of all which is also related to the data problem. Developed algorithms mostly include a combination of wavelet transform [14–19,9], discrete Fourier transform [7,21] or discrete cosine transform [21] and a support vector machine or artificial neural network classifier. B-scans keep the geometrical coherence of the defect, as can be seen in Fig. 1, which leads to a better noise invariance [24]. However, the analysis of B-scans can only be seen in a few works [5,6]. In [5] two popular deep learning object detection models, YOLOv3 [28] and SSD [29], have been used for defect detection. In [6] a deep learning classifier has been tested on augmented images, but with only three defects in the specimen block. Regarding C-scans, in [26] a method based on the comparison of the scan with a reconstructed reference image has been made. The method was able to detect all defects in their dataset, but with a high number of false-positive detections. There have also been some attempts in estimating defects from noisy measurements using Bayesian analysis [27].

There have been some attempts in using data augmentation to enlarge existing datasets. As mentioned, in [6], although only three defects were present in the test block, a copy/pasting data augmentation has been used to enlarge the dataset for training a deep learning detector. There are many variations on pasting and blending objects on the background in order to make the images look as realistic as possible. For instance, it can be done using Gaussian blur or Poisson blending [30] to smooth the edges. In [31] a comparison between different merging techniques has been made, using a combination of blending methods performed the best for most objects. On the other hand in [32] authors have pasted objects on random backgrounds and achieved improvements without any blending. Finally, generative adversarial networks (GANs) have recently become a popular choice for synthetic data generation and augmentation. GANs were first conceptualized in [33] in 2014. They can be used to generate images, video, audio, text, and much more. The development of the GAN came a long way in a short period of time. There are many different GAN architectures. Interesting approaches to GANs are image-to-image translation models. They are used for style transfer between images [34], image inpainting [35] and even generating images from masks [36]. One of the examples of those models is the Pix2pixGAN [36] and its successor pix2pixHD [37]. GANs show promising results in generating realistic images for human faces from noise with StyleGan2 [38] or converting position mask images to street-view with Pix2pixHD. A lot of work has been done for enlarging data sets in medical imagery. Pix2pixHD has proved to be useful in generating skin lesion images using semantic label maps [39]. An Inception-v4-based classifier [40] has been trained using real and combined real and data generated with the Pix2pixHD. Training the classifier on a combined real and generated data achieved a 1% improvement of the area under the ROC curve. In [13] authors have applied the GAN framework to synthesize high-quality liver lesion images for improved classification. In [41] authors have developed a multi-channel GAN (M-GAN) to generate PET images from CT scans. A similar approach with a cGAN has been made in [42,43]. Using generated data, they have achieved a 28% reduction in average false positive per case. Generating MR images from CT scans with paired and unpaired data has been researched in [44]. An MR-GAN with a concept inspired by CycleGAN [34] has been developed for this purpose. In [45] a DCGAN has been employed to generate realistic brain MR images. Data augmentation using non-convolutional GAN has been tested on three different non-image datasets [46]. Generated data has performed even better than real data when classifying using a Decision Tree (DT) classifier.

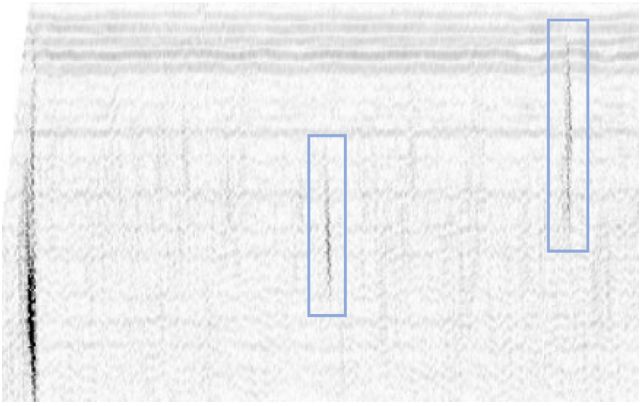


Fig. 1. Example of an ultrasonic B-scan with defects. The defects are indicated by bounding boxes.

1.3. Outline

The rest of the paper is organized as follows. Section 2 gives a detailed description of the used dataset. Section 3 describes the experimental procedure. Proposed GAN architecture and copy/-pasting method are presented in Section 4. Results are shown in Section 5 follow by the conclusion in Section 6.

2. Dataset

The dataset was obtained by scanning six steel blocks containing artificially created defects in the internal structure. Blocks varied in size and contained between six to 34 defects. In total there were 68 defects. Blocks were scanned using INETEC Dolphin scanner with a phased array probe. An INETEC phased array ultrasound transducer with a central frequency of 2.25 MHz was used. Angles ranging from 45 degrees up to 79 degrees with a 2-degree increment were acquired during the scanning. Blocks were also scanned with a skew of zero and 180 degrees. INETEC SignyOne data acquisition and analysis software was used to process the data and create B-scans (further noted as images) that were used in the dataset. Data were converted to B-scans as-is, without pseudo-coloring, as grayscale images. All images were converted into patches of size 256x256 pixels and annotated by multiple human experts. There were in total 3825 images with a total of 6238 annotations. We split the dataset into subsets for training, validation and testing. Details of the train, validation and test subsets can be seen in Table 1. Each subset contains unique defects that do not appear in other subsets. Our dataset is highly realistic and finding all of the defects is challenging even for the human inspectors. As for the copy/pasting method, we copied the defects from the training subset in the form of the rectangle patches from the annotations and pasted them on the empty UT backgrounds. There were 3400 empty UT backgrounds from all of the blocks and 4283 defect patches from the training subset.

3. Synthetic data generation

The acquired dataset is not large enough to properly train an object detector to detect defects. For this reason, we propose two methods to expand our dataset with synthetic data.

In this section, we have described the procedure of the experiment in this work. We developed two methods for synthetic image generation and use a state-of-the-art object detector for defect detection to test the quality of the generated data. We start by describing the current state-of-the-art method for generating images and proceed to describe a deep learning approach with our GAN. We then explain the usage of the object detector in the experiment.

Our first generative method is a copy/pasting (C/P) technique. Copy/pasting is a very logical method for enlarging the ultrasonic dataset because of the large number of B-scans without defects. We call these images canvases because we paste extracted defects on them. We extracted all of the defects from the training set and pasted them on canvases in random locations. The exact method is explained in the next section. An example of an empty image canvas, extracted defect, its pseudo mask, and the resulting image can be seen in Fig. 2.

The second method we propose is our own GAN architecture for the purpose of generating UT B-scans. Our GAN is an image-to-image GAN. This means that the position mask used as the network's input is translated to a realistic B-scan with defects at specified positions. We make position masks from all annotated images in the training set. An example of an input–output pair is shown in Fig. 3. Position masks on the input of the generator serve as a location label for the desired position of the defect on the generated image. The main novelty of our GAN is the usage of a pre-trained object detector for training the GAN. We use the object detector as an additional discriminator to provide information on the quality of the defect on the generated image when compared to the real image. It is important that the defect is positioned accurately as drawn in the position mask and that it is merged well with the background. After training the GAN we generated new position masks used for the generation of synthetic data. We determine the sizes and shapes of the defects on the position masks by extracting the aspect ratios of all annotations from the training set. Our generated images contain between one and four defects per image.

To estimate the quality of the generated images we used a popular object detector YOLOv3 [28]. This detector was already proven to work well for the task of defect detection from UT images in [5]. It is currently the state-of-the-art in defect detection. We first trained the detector using only real images and some traditional augmentations explained in the next section. We then tried training the object detector with images generated using the copy/paste method. We also trained the detector with a combination of real and generated images. Finally, we generated synthetic data with our GAN and again trained the object detector with generated images and a combination of real and generated data. Each of the trained versions of the object detector was tested on the same test dataset described in the previous section. Also, the same validation set was used in all three training variations.

Table 1
Number of images and annotations in train, validation and test subsets

	TRAIN	VALIDATION	TEST
Number of images	2278	379	1168
Number of annotations	4283	745	1210

4. Methods

In this section, a detailed explanation of developed methods is given. First, the copy/pasting method is described. Then the architecture of our proposed GAN is described with all of its special features. In the end, a short overview of the used object detector is given.

4.1. Copy/paste method

We used copy/paste method as a baseline to illustrate the complexity of generating synthetic data. While these images might look visually appealing, they are not of the same quality as the ones generated by the GAN.

As mentioned in Section 2 we have previously extracted defects from images in the training set. We paste them on random locations on images without visible defects. The process goes as follows. First, we randomly pick a canvas and randomly select the defect that will be pasted on it. We then put a threshold on a defect image. We make a binary pseudo mask by creating a binary image from the thresholded image and dilate it for two iterations with a 5x5 kernel. We then use the mask to extract only the defect from the initial defect patch image. We randomly select the position where we will paste the defect and calculate the compatibility of the selected defect background and the canvas on that location. We calculate the compatibility by calculating an intensity value of the background of the canvas and the defect. If these two values do not differ by more than 5%, we accept the proposed location. If these two values differ by more than 5% we try to select another location. We then select another image/canvas pair and repeat the process. For each new image, we set the limit of 100 attempts after which we just move on to generate another image. Usually, this limit is rarely reached since the right pair of canvas/defect nad location is usually found quickly. When the right pair is found, we proceed to paste the defect on the canvas. We first adapt the brightness of the defect to even further match the one from the canvas. We calculate the brightness of the location on the

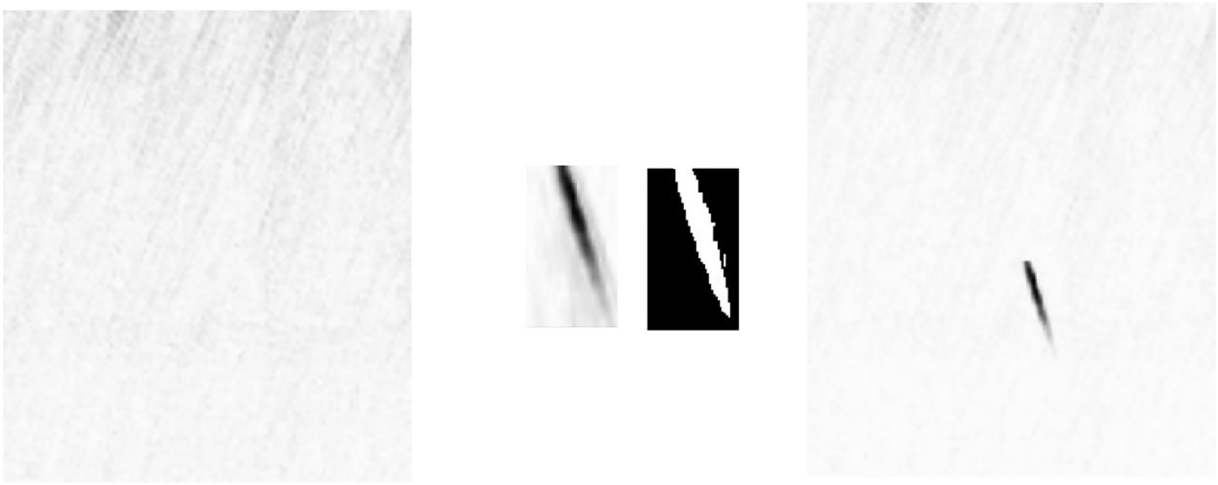


Fig. 2. Example of (from left to right) an empty canvas (B-scan without defects), an extracted defect and its binary pseudo mask, and resulting generated image.

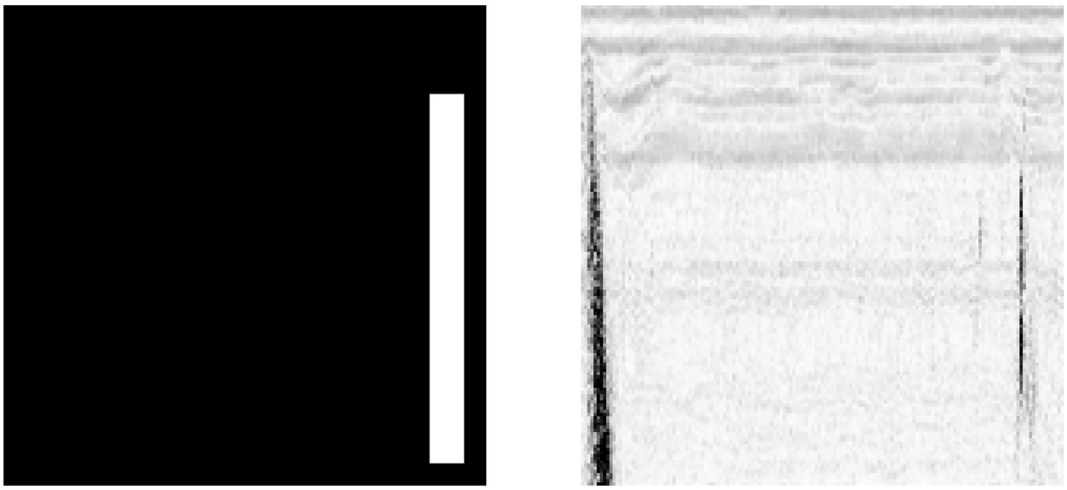


Fig. 3. Example of GAN input position mask (left) and corresponding desired output with the defect (right).

canvas and adapt the brightness of the defect to it. All that is left after that is to paste the defect. We concatenate the canvas and the defect, calculate the per pixel minimum of the two and merge it into one resulting image.

Samples of the real image, image generated with our GAN, and image generated with a copy/paste method are shown in Fig. 5.

4.2. GAN

The basic architecture of GAN is a combination of two neural networks, a generator, and a discriminator. The generator generates high-quality images from random noise. Noisy input helps generate a wide selection of images from the learnt distribution. Discriminator on the other hand tries to distinguish between generated images and the real ones. The constant rivalry between the generator and the discriminator is what makes GANs adversarial. Mathematically, discriminator and generator play a minimax game with the following function [47]:

$$\min_G \max_D V(D,G) = E_s[log(D(s))] + E_z[1 - log(D(G(z)))] \quad (1)$$

where : G = the generator
D = the discriminator
s = training sample
z = random variable

The goal of the generator is to maximize the probability of discriminator labeling generated images as real samples and the discriminator has the goal of minimizing that probability while being able to label real data as such. This neural network configuration enables unsupervised learning of both generator and discriminator. For image generating purposes it is convenient to use convolution operations in GAN which is presented in Deep Convolutional GAN (DCGAN) [48].

We call our GAN the DetectionGAN (DetGAN) for its specific architecture. We base it on the Pix2pixHD implementing some of the features from it. Our DetectionGAN consists of a U-net generator with skip connections, two PatchGAN discriminators [36] that work on different scales and a pre-trained object detector which serves as an additional discriminator. We train the proposed GAN with image pairs of real images and their position masks. Position masks can be viewed as a conditional input of the generator and the discriminator. This version of GAN is called a conditional GAN and its objective can be express as:

$$\min_G \max_D V(D,G) = E_{x)}[log(D(x,s))] + E_z[1 - log(D(x,G(x,z)))] \quad (2)$$

where: x = conditional variable

Input to the generator is the position mask defining the position of the defect. Proposed GAN does not have an input noise. The output of the generator is connected to the discriminators and the object detector. There is a total of 54,409,603 parameters in the generator. All of them are randomly initialized and trained. Unlike in Pix2pixHD, we do not use a two-stage generator nor do we upscale the position mask to generate a higher resolution image. We use skip connections with concatenation in the generator.

Discriminator has a position mask concatenated to the generated or real image as an input. Image and mask are concatenated across the channels axis. The goal of the concatenation of the position mask and the image is to provide information on the position of defects in the image for the discriminator. This concatenation leads to an improvement as shown in Section 5. Discriminator gets the real and the generated image during each step as an input. In order to discriminate images on two different scales, we use two discriminators. This way we can generate more realistic images with both coarse and fine details. Both discriminators have 1,391,554 parameters that are randomly initialized.

For the additional discriminator we use a YOLO object detector during this experiment, but any other object detector could be used. Usage of the YOLO discriminator helps the GAN with the generation of highly realistic images with defects in precise, desired locations. We input the generated image and then the real image and compare the outputs. We want these two outputs to be the same so that there is no difference between the generated and real image for the detector. This way we ensure defects are placed on the exact locations and without any artifacts. Using an object detector as a discriminator provides a significant improvement as shown in Section 5. To the best of our knowledge, this is the first time an object detector has been used as a discriminator in a GAN in order to enhance the quality of generated images.

An illustration of the proposed GAN is shown in Fig. 4. Filter sizes of each layer of the generator and discriminators are noted in the figure. Overall, the forward pass of our model can be explained as:

$$\begin{aligned} &G(x) = g \\ &D_1(concatenate(x,r)) = d_{11}, fm_{11} \\ &D_1(concatenate(x,g)) = d_{12}, fm_{12} \\ &D_2(downsample(concatenate(x,r))) = d_{21}, fm_{21} \\ &D_2(downsample(concatenate(x,g))) = d_{22}, fm_{22} \\ &Y(upsample(r)) = y_1 \\ &Y(upsample(g)) = y_2 \end{aligned} \quad (3)$$

where :
- G = the generator
- D_1 = the discriminator 1
- D_2 = the discriminator 2
- Y = the YOLO discriminator
- g = generated image
- x = positional mask
- r = real image
- d_{ij} = output of the discriminator
- fm_{ij} = second to last layer of discriminator
- $downsample()$ = downsampling by a factor of 2
- $upsample()$ = upsampling to 416x416 px

We train our GAN using a set of loss functions. For the generator, we use four different losses. At the output of the generator, we calculate the L1 loss on the generated image and the paired real image:

$$G_{gloss} = |g - r| \quad (4)$$

For propagating discriminator output to the generator we use the mean squared error loss:

$$G_{dloss} = \frac{1}{2}\sum_{k=1}^{2}(d_{k1} - d_{k2})^2 \quad (5)$$

We also use the feature matching loss for training the generator, similar to the one in [37]:

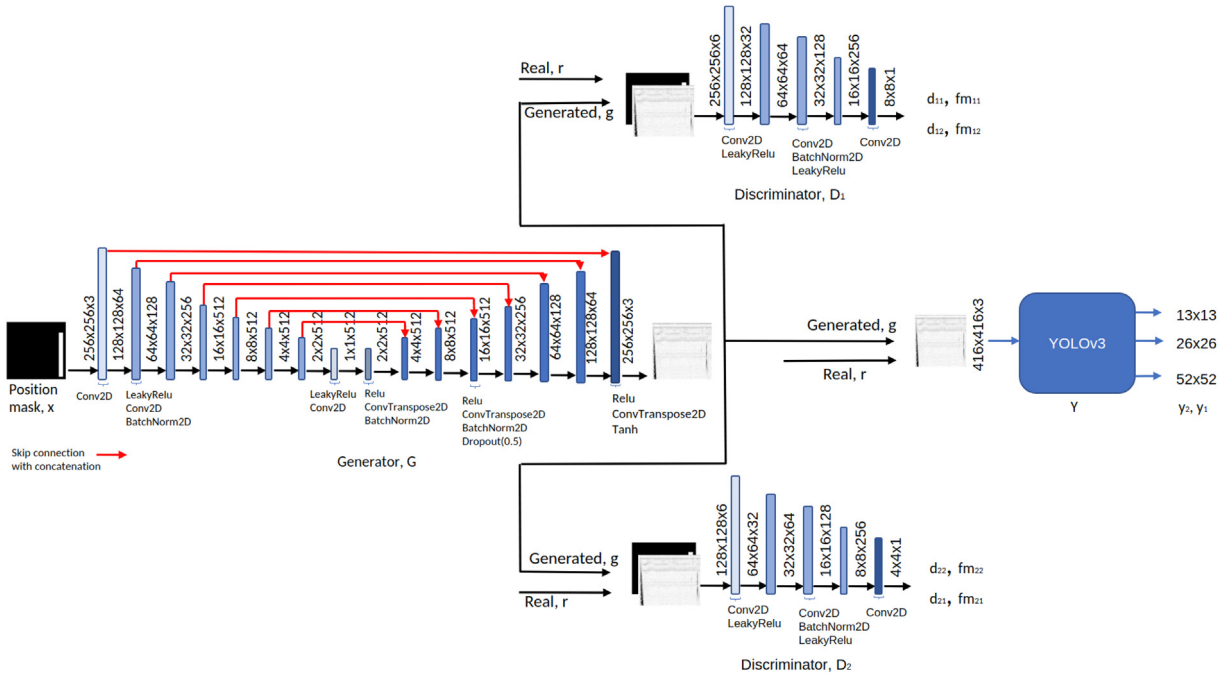


Fig. 4. Simplified architecture of our DetectionGAN.

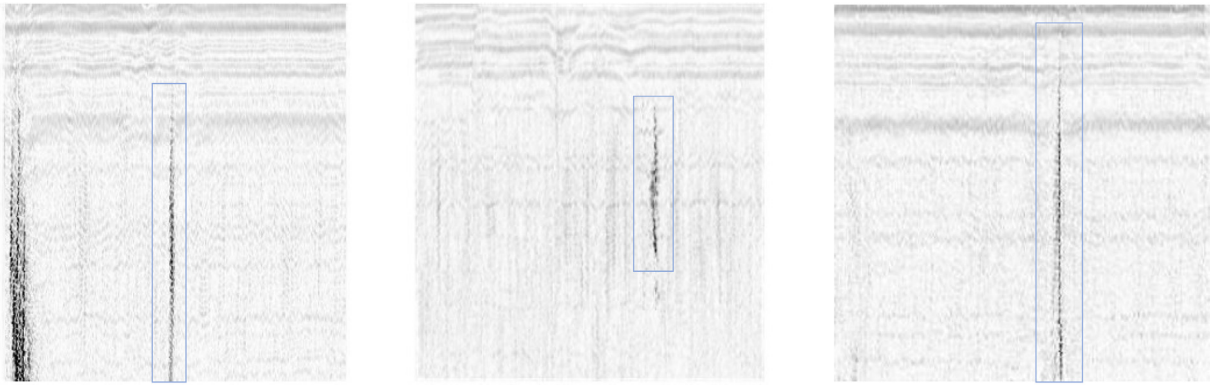


Fig. 5. Samples of (from left to right): real image, an image generated with copy/paste method, an image generated with DetectionGAN. In each image, the defects are indicated by a bounding box.

$$G_{fmloss} = \sum_{k=1}^{2} \frac{1}{2} |fm_{k1} - fm_{k2}| \tag{6}$$

We also compare the output of all three scales of the object detector when inputting the real image and the generated one. We again use the L1 loss to propagate the error down to the generator:

$$G_{Yloss} = |y_1 - y_2| \tag{7}$$

For training the discriminator we use the mean square error loss, just like it is used in Pix2pixHD. Since we have two discriminators we have a multi-task learning problem of

$$\min_{G} \max_{D_1, D_2} \sum_{k=1,2} \mathscr{L}_{\text{GAN}}(G, D_k) \tag{8}$$

4.3. Object detector

Our object detector, You Only Look Once (YOLO) version 3 is taken from [5] where it proved to be able to detect defects with high average precision.

YOLOv3 is an object detector that belongs to the one-stage detector family. This means that the model directly searches for objects' presence at predefined places without performing a region proposal step. The detector consists of a backbone network, Darknet-53 [28], used to extract useful features from the image and the detection head. The detection head is used for the localization and classification of the objects. To improve invariability to objects sizes, the detection process is performed at three different scales using feature maps with resolutions: 13x13, 26x26, and 52x52. Each value of the feature map is used to perform three predictions (for objects of 3 different aspect ratios). The coordinates of the bounding boxes can then be determined by decoding the predictions. Non-maximum-suppression and object threshold are also performed after the decoding to limit the number of predicted bounding boxes and keep only the boxes that encapsulate the object the best. For the training of the GAN, we use outputs of the three mentioned feature maps from the YOLO.

The aim of this work is to improve the performance of the object detector using synthetic data. We train the object detector on real, generated data, and a combination of those two. We train the networks with the same hyperparameters in order to have a fair comparison. We first trained the detector on real data, tuned the hyperparameters to achieve the best possible performance and used the same hyparameters for training with other data combinations. We input images of size 416x416 pixels. We used a pretrained backbone and froze its parameters while training. Hence, we trained only 20,974,518 of a total number of 61,576,342 parameters.

5. Experimental setup and results

5.1. Experimental setup

In this section, we describe the experiment and hyparameters used to train our GAN and the object detector. Our experiment goes as follows. We first trained an object detector with real data. This trained network is used as the YOLO discriminator of our GAN. We generated synthetic images using the copy/pasting method and the GAN method. We generated 200,000 synthetic images with both the copy/paste and DetectionGAN method. We also trained the DetectionGAN without using the object detector discriminator and concatenation in the discriminators to compare the effectiveness of each proposed modification. For each version of the proposed GAN, we use the same position masks to generate synthetic images. We again train the object detector using the generated data and compare results.

For training the object detector we used the following configuration. We use batch size eight and Adam optimizer with a learning rate of 1e-3. Anchor hyperparameters were calculated using the K-means with Jaccard distance as proposed in [49]. Custom anchors were calculated on the training set for all of the training combinations. We slightly changed only the ignore threshold hyperparameter to 0.6 from the original YOLO implementation. We used checkpoints while training the model. An early stopping callback was used to stop the training after the validation loss didn't improve for over eight epochs. We reduced the learning rate after every two epochs with no improvement on the validation set. We also used some basic augmentations while training all of the models. Those augmentations include horizontal image flipping, random cropping, and HSV space modulation. We also tried training the object detector without augmentations. It took us around 30 min to train the object detector. For testing the object detector we used the following hyperparameters. The object threshold of YOLO was 0.001 while the non-maximum suppression threshold was 0.5 and the intersection over union threshold was 0.5. These hyperparameters are a standard for evaluating object detection challenges and were used in [5].

We train our GAN as follows. Position masks for the input of the GAN are of size 256x256 pixels, as well as the generated images. For training the generator we use an Adam optimizer with a first-moment term of 0.5, the second one of 0.999, and a learning rate of 0.0002. One of our discriminators has an input image of 256x256 pixels, while the other one has a downscaled image of

128x128 pixels as an input. We train discriminators using Adam optimizer with the same parameters as the generator.

We use the pre-trained object detector and do not train it during training the GAN. We implement a set of simple data augmentations for training the GAN including horizontal flipping, brightness modulation, and random cropping. These data augmentations enable us to achieve great results in training the GAN and generating high-quality images. With data augmentation, we expand the training subset to over 9000 images. We train the GAN for 800 epochs and for the last 100 epochs we linearly reduce the learning rate to zero. Each epoch corresponds to one pass through all the images in the training dataset. We trained with a batch size eight. The training takes around 96 h using a single NVIDIA RTX 2080Ti graphics card. Although this may seem like a long time it is similar to the training time of pix2pix [36]. Taking into consideration the number of images in the training subset, the number of epochs, and the complexity of the whole GAN it is the expected training duration.

5.2. Results and discussion

The performance of the proposed approach was tested on a test subset. As described in Section 3 we test the quality of generated images by training an object detector on real and generated images. We used an average precision (AP) metric for assessing the performance of an object detector on a test set. Each experiment with the object detector was run three times. In tables, we present the mean value and the standard deviation for each result. Generated images used in this test were not handpicked but randomly generated.

Detailed results can be seen in Table 2. We ran the training of the object detector on real data with and without data augmentation. Using the C/P method for image synthesis did not provide any improvements in the detection. When training only on C/P images we acquire a result of only 51% AP on the same test set. When training on the combination of both real and C/P images we again do not get any improvements. The reason could be that this data has some artifacts when compared to the real images. Although visually, both images generated with DetectionGAN and with the copy/paste method look realistic, the object detector tends to learn wrong features and can not converge to a better model than the one trained on real images. However, we achieved an improvement with DetectionGAN-generated images when opposed to training the object detector with only real images. An improvement of 2% has been achieved while training only on DetectionGAN-generated images, and an improvement of almost 6% of AP was achieved when training on a combined dataset of real and images generated with our GAN. As a reference, experiments with two versions of the DetectionGAN without the object detector discriminator and without position mask and image concatenation in the discriminator were made. Both versions perform worse than our DetectionGAN.

Table 2
Results of training the object detector on different training datasets

TRAINING DATA	AP (%)	# IMAGES
Real w/ augm.	70.36 σ 1.11	2,278
Real w/o augm.	47.64 σ 6.16	2,278
C/P	51.13 σ 2.72	200,000
C/P + real	68.07 σ 1.57	202,278
DetGAN w/o conc.	40.78 σ 2.35	200,000
DetGAN w/o conc. + real	67.78 σ 3.14	202,278
DetGAN w/o yolo	62.60 σ 0.85	200,000
DetGAN w/o yolo + real	69.40 σ 2.83	202,278
DetectionGAN	72.17 σ 0.16	200,000
DetectionGAN + real	**75.91** σ 0.95	202,278

The final score for defect detection is almost 76%, which seems rather low, but this is due to a very difficult dataset used, which is problematic even for human inspectors. Such problematic datasets are the primary target for result improvements. The presented results demonstrate that we have obtained our initial goal of improving the detection results using realistic synthetic samples produced by our generation method. The results also show that the existing generative methods are not capable of generating images of sufficient quality to improve the performance of the defect detector.

This experiment indicates that it is important to have the most realistic data as it is possible to achieve an improvement. We illustrated the complexity of the problem of generating synthetic data for training the object detector. Our proposed GAN can generate highly realistic data that can improve the object detector's performance.

6. Conclusion and future work

In this paper, we propose a novel generative adversarial network for generating highly realistic B-scans (images) from position mask images. Our DetectionGAN generates highly realistic ultrasonic images from position masks that can be used to train an object detector. We achieved an improvement of almost 6% while training on a combination of generated and real data. We also developed a copy/pasting method based on the previous state-of-the-art approach for data augmentation to compare it to our proposed method. As we didn't cherry-pick DetectionGAN-generated images all of the generated images were proven to be of high quality.

With the increasing problem of lack of data and advances in generating high-quality synthetic data, networks such as our DetectionGAN could be used in many science and industry fields. In future work, DetectionGAN should be tested using different object detectors as discriminators. Also, other state-of-the-art object detectors should be tested on this dataset.

CRediT authorship contribution statement

Luka Posilović: Conceptualization, Methodology, Software, Validation, Data curation, Writing - original draft. **Duje Medak:** Software, Data curation, Writing - review & editing. **Marko Subašić:** Conceptualization, Resources, Writing - review & editing, Supervision. **Marko Budimir:** Resources, Data curation, Writing - review & editing, Funding acquisition. **Sven Lončarić:** Resources, Writing - review & editing, Supervision, Funding acquisition.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research was co-funded by the European Union through the European Regional Development Fund, under the grant KK.01.2.1.01.0151 (Smart UTX).

References

[1] L. Cartz, Nondestructive testing: radiography, ultrasonics, liquid penetrant, magnetic particle, eddy current, ASM International (1995), url: https://books.google.hr/books?id=0spRAAAAMAAJ.

[2] J. Ye, S. Ito, N. Toyama, Computerized ultrasonic imaging inspection: From shallow to deep learning, Sensors 18 (11) (2018) 3820, https://doi.org/10.3390/s18113820.
[3] P. Broberg, Imaging and analysis methods for automated weld inspection, Ph. D. thesis, Luleå tekniska universitet (2014)..
[4] J. Krautkrämer, H. Krautkrämer, Ultrasonic Testing of Materials, Springer-Verlag (1983), url:https://books.google.hr/books?id=AvwrAAAAIAAJ.
[5] L. Posilović, D. Medak, M. Subašić, T. Petković, M. Budimir, S. Lončarić, Flaw detection from ultrasonic images using yolo and ssd, in: 2019 11th International Symposium on Image and Signal Processing and Analysis (ISPA), IEEE, 2019, pp. 163–168.
[6] I. Virkkunen, T. Koskinen, O. Jessen-Juhler, J. Rinta-Aho, Augmented ultrasonic data for machine learning, Journal of Nondestructive Evaluation 40 (1) (2021) 1–11.
[7] I.S. Souza, M.C. Albuquerque, E.F. de SIMAS FILHO, C.T. FARIAS, Signal processing techniques for ultrasound automatic identification of flaws in steel welded joints–a comparative analysis, in: 18th World Conference on Nondestructive Testing, 2012, pp. 16–20..
[8] N. Munir, H.-J. Kim, S.-J. Song, S.-S. Kang, Investigation of deep neural network with drop out for ultrasonic flaw classification in weldments, Journal of Mechanical Science and Technology 32 (7) (2018) 3073–3080, https://doi.org/10.1007/s12206-018-0610-1.
[9] M. Meng, Y.J. Chua, E. Wouterson, C.P.K. Ong, Ultrasonic signal classification and imaging system for composite materials via deep convolutional neural networks, Neurocomputing 257 (2017) 128–135.
[10] C. Shorten, T.M. Khoshgoftaar, A survey on image data augmentation for deep learning, Journal of Big Data 6 (1) (2019) 60.
[11] D. Ventura, S. Warnick, A theoretical foundation for inductive transfer, Brigham Young University, College of Physical and Mathematical Sciences..
[12] D. Soekhoe, P. Van Der Putten, A. Plaat, On the impact of data set size in transfer learning using deep neural networks, in: International Symposium on Intelligent Data Analysis, Springer, 2016, pp. 50–60.
[13] M. Frid-Adar, I. Diamant, E. Klang, M. Amitai, J. Goldberger, H. Greenspan, Gan-based synthetic medical image augmentation for increased cnn performance in liver lesion classification, Neurocomputing 321 (2018) 321–331.
[14] F. Bettayeb, T. Rachedi, H. Benbartaoui, An improved automated ultrasonic nde system by wavelet and neuron networks, Ultrasonics 42 (1) (2004) 853–858, Proceedings of Ultrasonics International 2003..
[15] S. Sambath, P. Nagaraj, N. Selvakumar, Automatic defect classification in ultrasonic ndt using artificial intelligence, Journal of Nondestructive Evaluation 30 (1) (2011) 20–28.
[16] Y. Chen, H.-W. Ma, G.-M. Zhang, A support vector machine approach for classification of welding defects from ultrasonic signals, Nondestructive Testing and Evaluation 29 (3) (2014) 243–254. arXiv:https://doi.org/10.1080/10589759.2014.914210, doi:10.1080/10589759.2014.914210..
[17] A. Al-Ataby, W. Al-Nuaimy, C. Brett, O. Zahran, Automatic detection and classification of weld flaws in tofd data using wavelet transform and support vector machines, Insight – Non-Destructive Testing and Condition Monitoring 52 (2010) 597–602, https://doi.org/10.1784/insi.2010.52.11.597.
[18] V. Matz, M. Kreidl, R. Smid, Classification of ultrasonic signals, International, Journal of Materials 27 (2006) 145, https://doi.org/10.1504/IJMPT.2006.011267.
[19] M. Khelil, M. Boudraa, A. Kechida, R. Drai, Classification of Defects by the SVM Method and the Principal Component Analysis (PCA) (09 2007). doi:10.5281/zenodo.1060751..
[20] M. Meng, Y.J. Chua, E. Wouterson, C.P.K. Ong, Ultrasonic signal classification and imaging system for composite materials via deep convolutional neural networks, Neurocomputing 257 (2017) 128–135, machine Learning and Signal Processing for Big Multimedia Analysis. doi: 10.1016/j.neucom.2016.11.066..
[21] F. Cruz, E.S. Filho, M. Albuquerque, I. Silva, C. Farias, L. Gouvêa, Efficient feature selection for neural network based detection of flaws in steel welded joints using ultrasound testing, Ultrasonics 73 (2017) 1–8, https://doi.org/10.1016/j.ultras.2016.08.017.
[22] G.A. Guarneri, F.N. Junior, L. de Arruda, Weld discontinuities classification using principal component analysis and support vector machine, XI Simpósio Brasileiro de Automaçao Inteligente (2013) 2358–4483.
[23] J. Veiga, A.A. de Carvalho, I. Silva, J.M.A. Rebello, The use of artificial neural network in the classification of pulse-echo and tofd ultra-sonic signals, Journal of The Brazilian Society of Mechanical Sciences and Engineering – J BRAZ SOC MECH SCI ENG 27. doi:10.1590/S1678-58782005000400007..
[24] H. Cygan, L. Girardi, P. Aknin, P. Simard, B-scan ultrasonic image analysis for internal rail defect detection, in: World Congress on Railway Research, 2003..
[25] A. Kechida, R. Drai, A. Guessoum, Texture analysis for flaw detection in ultrasonic images, Journal of Nondestructive Evaluation 31 (2) (2012) 108–116, https://doi.org/10.1007/s10921-011-0126-4.
[26] H. Kieckhoefer, J. Baan, A. Mast, W.A. Volker, Image processing techniques for ultrasonic inspection, in: Proc. 17th World Conference on Nondestructive Testing, Shanghai, China, 2008..
[27] A. Dogandzic, B. Zhang, Bayesian nde defect signal analysis, Signal Processing, IEEE Transactions on 55 (2007) 372–378, https://doi.org/10.1109/TSP.2006.882064.
[28] J. Redmon, A. Farhadi, Yolov3: An incremental improvementCite arxiv:1804.02767Comment: Tech Report. url:https://arxiv.org/abs/1804.02767.
[29] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S.E. Reed, C. Fu, A.C. Berg, SSD: single shot multibox detector, CoRR abs/1512.02325. arXiv:1512.02325, doi:10.1007/978-3-319-46448-0_2..
[30] P. Pérez, M. Gangnet, A. Blake, Poisson image editing, ACM SIGGRAPH 2003 Papers (2003) 313–318.
[31] D. Dwibedi, I. Misra, M. Hebert, Cut, paste and learn: Surprisingly easy synthesis for instance detection, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 1301–1310.
[32] J. Rao, J. Zhang, Cut and paste: Generate artificial labels for object detection, in: Proceedings of the International Conference on Video and Image Processing, 2017, pp. 29–33.
[33] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Advances in Neural Information Processing Systems, 2014, pp. 2672–2680..
[34] J.-Y. Zhu, T. Park, P. Isola, A.A. Efros, Unpaired image-to-image translation using cycle-consistent adversarial networks, in, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2223–2232.
[35] G. Liu, F.A. Reda, K.J. Shih, T.-C. Wang, A. Tao, B. Catanzaro, Image inpainting for irregular holes using partial convolutions, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 85–100.
[36] P. Isola, J.-Y. Zhu, T. Zhou, A.A. Efros, Image-to-image translation with conditional adversarial networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1125–1134.
[37] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, B. Catanzaro, High-resolution image synthesis and semantic manipulation with conditional gans, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 8798–8807.
[38] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, T. Aila, Analyzing and improving the image quality of StyleGAN, in: Proc. CVPR, 2020..
[39] A. Bissoto, F. Perez, E. Valle, S. Avila, Skin lesion synthesis with generative adversarial networks, in: OR 2.0 Context-Aware Operating Theaters, Computer Assisted Robotic Endoscopy, Clinical Image-Based Procedures, and Skin Image Analysis, Springer, 2018, pp. 294–302..
[40] C. Szegedy, S. Ioffe, V. Vanhoucke, A. Alemi, Inception-v4, inception-resnet and the impact of residual connections on learning, arXiv preprint arXiv:1602.07261..
[41] L. Bi, J. Kim, A. Kumar, D. Feng, M. Fulham, Synthesis of positron emission tomography (pet) images via multi-channel generative adversarial networks (gans), in: molecular imaging, reconstruction and analysis of moving body organs, and stroke imaging and treatment, Springer, 2017, pp. 43–51..
[42] A. Ben-Cohen, E. Klang, S.P. Raskin, M.M. Amitai, H. Greenspan, Virtual pet images from ct data using deep convolutional networks: initial results, in: International Workshop on Simulation and Synthesis in Medical Imaging, Springer, 2017, pp. 49–57.
[43] A. Ben-Cohen, E. Klang, S.P. Raskin, S. Soffer, S. Ben-Haim, E. Konen, M.M. Amitai, H. Greenspan, Cross-modality synthesis from ct to pet using fcn and gan networks for improved automated lesion detection, Engineering Applications of Artificial Intelligence 78 (2019) 186–194.
[44] C.-B. Jin, H. Kim, M. Liu, W. Jung, S. Joo, E. Park, Y.S. Ahn, I.H. Han, J.I. Lee, X. Cui, Deep ct to mr synthesis using paired and unpaired data, Sensors 19 (10) (2019) 2361.
[45] K. Kazuhiro, R.A. Werner, F. Toriumi, M.S. Javadi, M.G. Pomper, L.B. Solnes, F. Verde, T. Higuchi, S.P. Rowe, Generative adversarial networks for the creation of realistic artificial brain magnetic resonance images, Tomography 4 (4) (2018) 159.
[46] F.H.K. d. S. Tanaka, C. Aranha, Data augmentation using gans, arXiv preprint arXiv:1904.09135..
[47] S. Kazeminia, C. Baur, A. Kuijper, B. van Ginneken, N. Navab, S. Albarqouni, A. Mukhopadhyay, Gans for medical image analysis, arXiv preprint arXiv:1809.06222..
[48] A. Radford, L. Metz, S. Chintala, Unsupervised representation learning with deep convolutional generative adversarial networks, arXiv preprint arXiv:1511.06434..
[49] J. Redmon, A. Farhadi, Yolo9000: Better, faster, stronger, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 6517–6525, https://doi.org/10.1109/CVPR.2017.690.

Luka Posilović received his M.Sc. from the University of Zagreb, Faculty of Electrical Engineering and Computing in 2019. He is currently working as a young researcher in an Image Processing Group in the Department of Electronic Systems and Information Processing and working on his Ph.D. at the same University. His research interests include visual quality control, deep learning object detection, and synthetic image generation.

Duje Medak received his M.Sc. from the University of Zagreb, Faculty of Electrical Engineering and Computing in 2019. He is currently pursuing a Ph.D. degree in the same faculty while working as a researcher in the Image Processing Group in the Department of Electronic Systems and Information Processing. His research interests include image processing, image analysis, machine learning, and deep learning. His current research interest includes deep learning object detection methods and their application in the non-destructive testing (NDT) domain.

Marko Subašić is an associate professor at the Department for Electronic Systems and Information Processing, Faculty of Electrical Engineering and Computing, the University of Zagreb, and has been working there since 1999. He received his Ph.D. degree from the Faculty of Electrical Engineering and Computing, University of Zagreb, in 2007. His field of research is image processing and analysis and neural networks with a particular interest in image segmentation, detection techniques, and deep learning.

Marko Budimir received his M.Sc. of physics at the University of Zagreb, Faculty of Science in 2000., and his Ph.D. at Ecole Polytechnique Federale de Lausanne in Switzerland in 2006. He has worked at EPFL since 2006. till 2008. From 2008. he is working at the Institute of Nuclear Technology (INETEC). He coordinated many key projects at INETEC and although he is a key person in a company in the industry sector he is still working close to the field of science.

Dr. Sven Lončarić is a professor of electrical engineering and computer science at the Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia. As a Fulbright scholar, he received a Ph.D. degree in electrical engineering from the University of Cincinnati, OH in 1994. From 2001–2003, he was an assistant professor at the New Jersey Institute of Technology, USA. His areas of research interest are image processing and computer vision. He was the principal investigator on a number of R&D projects. Prof. Lončarić co-authored more than 250 publications in scientific journals and conferences. He is the director of the Center for Computer Vision at the University of Zagreb and the head of the Image Processing Group. He is a co-director of the Center of Excellence in Data Science and Cooperative Systems. Prof. Lončarić was the Chair of the IEEE Croatia Section. He is a senior member of IEEE and a member of the Croatian Academy of Technical Sciences. Prof. Lončarić received several awards for his scientific and professional work.

Publication 4

D. Medak, L. Posilović, M. Subašić, T. Petković, M. Budimir, S. Lončarić, ”Rapid Defect Detection by Merging Ultrasound B-scans from Different Scanning Angles”,in *Proc. of the 12th International Symposium on Image and Signal Processing and Analysis (ISPA)*, Sep. 2021, pp. 219-224.

Rapid Defect Detection by Merging Ultrasound B-scans from Different Scanning Angles

Duje Medak[1], Luka Posilović[1], Marko Subašić[1], Tomislav Petković[1], Marko Budimir[2], Sven Lončarić[1]
[1]University of Zagreb, Faculty of Electrical Engineering and Computing, Zagreb, Croatia
[2]INETEC Institute for Nuclear Technology, Zagreb, Croatia
Email: duje.medak@fer.hr

***Abstract*—Ultrasonic testing (UT) is a commonly used approach for inspection of material and defect detection without causing harm to the inspected component. To improve the reliability of defect detection, the material is often scanned from various angles leading to an immense amount of data that needs to be analyzed. Some of the defects are only seen on B-scans taken from a particular angle so discarding some of the data would increase the risk of not detecting all of the defects. Recently there has been significant progress in the development of methods for automated defect analysis from the UT data. Using such methods the inspection can be performed quicker, but it is still necessary to inspect all of the angles to detect defects. In this work, we test a novel approach for accelerating the analysis by merging the images from various angles. To reduce the information loss during the process of merging, we develop a new model with a weighting module that dynamically determines the importance of each of the scanning angles. Using the proposed module, the loss of information is minimal, so the precision of the detection model is comparable to the model tested on each of the images separately. Using the merged images input, the analysis can be accelerated by almost 15 times.**

***Index Terms*—image processing, image analysis, convolutional neural networks, ultrasonic imaging, nondestructive testing, automated flaw detection**

I. Introduction

Ensuring the safety and proper functioning of a system includes continuous monitoring of its critical parts. The flaws can appear inside the material due to harsh working conditions such as high temperature, or direct material stressing. Types and sizes of flaws can vary, and some of the material imperfections do not actually pose a safety threat or influence the system's proper functioning. However, a method that can detect even the slightest imperfection is desirable because it can be used to monitor critical parts and keep track of previously found flaws. This can help with the maintenance planning and proper timing of the replacement of the components which is an economically better option than the early retirement of the perfectly functioning and safe parts.

Ultrasonic testing (UT) is a widely used Non-destructive evaluation (NDE) technique for the inspection of material and flaw detection [1], [2]. This method is fairly simple to employ and allows a precise defect localization [2], [3]. This NDE technique uses an ultrasonic probe both as the transmitter and the receiver of the ultrasonic waves. The ultrasonic waves are propagated through the material until some change in the material density is encountered. When this happens, a fraction of the waves will be reflected and this will be picked up by the probe. The amount of received energy can be plotted as a function of material depth which allows the precise extraction of the flaw's location. This view of the acquired data is called an A-scan. One A-scan is obtained for each combination of the probe's position and angle of the transmitted ultrasonic waves. A sequence of A-scans will be obtained when the probe is moved along the surface of a material. A sequence of A-scans can be used to form an image representation called B-scan. This is done by transferring each A-scan into one image column by converting the amplitude of the A-scan into pixel values. Higher amplitudes are usually converted into darker pixel values so if some defect is present in the material it will be shown as a dark shape inside the image. Material is usually inspected using a special type of probe called a phased array probe. This probe allows simultaneous inspection of the material at various angles. An illustration of such probe is shown in Figure 1. As shown in the illustration, by moving the probe from one side of the material to the other several images will be obtained (one for each angle). The number of acquired images depends on the initial angle, final angle, and the increment. Defects look different when acquired at different angles, and sometimes a defect can not even be detected unless a convenient angle is chosen. This is why in real-life inspection, the procedures often require that all of the data, that was acquired at various angles, must be analyzed. This also means the analysis lasts longer regardless of whether

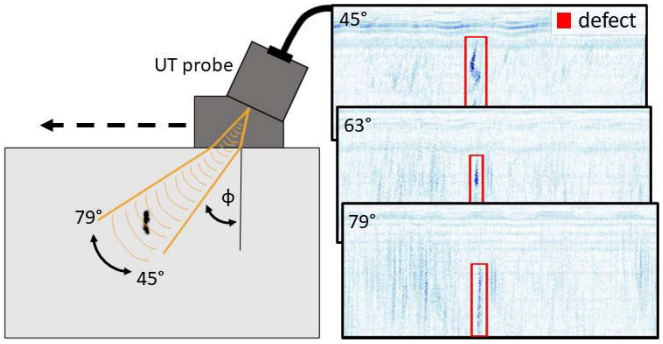


Fig. 1. Illustration of UT scanning with a phased array probe. Examples of B-scans that are obtained for various angle values are shown on the right side of the figure. Red bounding boxes mark the defect's location.

2021 12th International Symposium on Image and Signal Processing and Analysis (ISPA) | 978-1-6654-2639-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/ISPA52656.2021.9552050

the analysis is performed manually or by using some data analysis software.

An immense amount of data motivated the researchers to develop methods for automated UT data analysis. Developed methods work with different types of data (A-scans, B-scans, C-scans) as well as the other NDE techniques such as eddy current, radiography, and thermography to name a few. Early attempts of automated analysis of UT data relied on signal processing techniques. The most popular such approach is to calculate the coefficients from the A-scans using the wavelet transform or Fourier transform. The calculated coefficients can then be fed into a classifier such as support-vector machine (SVM) [4]–[6] or artificial neural network (ANN) [2], [7], [8]. In [9] the authors rearranged the calculated coefficients into a matrix and then used a special type of ANN called convolutional neural network (CNN). CNN's have been a dominant approach in computer vision tasks [10]–[12] for a long time and recently several authors demonstrated the merits of such architectures for UT data analysis [9], [13]. Due to increased availability of computing power and improved efficiency of the recent CNNs [14]–[16], the training of such models on a small dataset became feasible and yields good results when applied on NDE data analysis [17]–[21]. Even if the amount of available UT B-scans data is not sufficient some authors managed to develop solutions based on CNN by using the simulated [22] or artificially expanded (augmented) datasets [23]. To the best of our knowledge, there were no previous attempts of simultaneous analysis of UT images taken at different angles. All of the solutions for the automated analysis presented so far have to be run separately for each of the acquired angles which can be very time-consuming. We are trying to tackle this problem by developing a novel method for simultaneous analysis of B-scans taken at different angles.

II. Methodology

In this work, we propose a novel approach to speed up the analysis of the data by merging the images taken at various angles. To minimize the information loss during this process, we extended the used detection architecture with a module that performs image merging. We call the proposed module Angles Analysis Module (AAM). AAM dynamically determines which of the input images contain relevant information and gives higher importance to such images. The proposed module is trained jointly with the deep learning object detector that is used to localize defects. Using the AAM the defects' signals are more likely to remain in the resulting image while the noise from irrelevant images can be decreased. We experimentally confirmed that the proposed solution improves the results of defect detection compared to default image merging.

We start the development of our method by taking a well-established deep learning object detection architecture EfficientDet [25]. We use the smallest available model from the EfficientDet family called EfficientDet-D0. Additionally, we decreased the input image resolution size to 384x384 since most of our images are smaller than that. We trained and evaluated this model on a full dataset of volume corrected B-scans (VC-B-scans) that were taken at different angles. We then test several approaches for merging the information from many B-scans images into a new image that can be fed to the object detector. The merging process must be simple and fast in order to retain the advantage over the approach that analyzes the full dataset. To test the performance of a new model, we created a new dataset with appropriate bounding boxes. It is important that the new dataset still has labeled all of the defects. More details about the used datasets are given in section III-A.

The first approach that we tried was to simply merge all of the images with a minimum pooling along the angles axis while giving each of the input images equal importance. In the rest of this section, A denotes the number of different angles that were used during the acquisition of the images. The value for each of the resulting pixels can be calculated by finding the pixel among the A images that has the smallest value:

$$I_m(x,y,z) = \min_{\forall a \in \{0,1,\ldots,A-1\}} I(a,x,y,z) \tag{1}$$

As stated in the introduction, smaller pixel values represent a larger amount of reflected ultrasound energy. Performing the minimum pooling in the described way will highlight the defects' signals but it will also increase the noise. If some of the images from the sequence do not contain any defect, using them during the merging process can unnecessarily increase the noise. However, we do not know in advance which of the images are useful and which are not since the defects appear uniformly across all angles in the dataset. To solve this issue, we extend the standard object detector pipeline. The proposed approach is shown in Figure 2. Our model first takes a quick look at all of the angles using the Angles Analysis Module. The output of the module is a vector of weights W that determines the importance for each of the images from the input sequence. Since the images are passed through minimum pooling operation after the weighting, the images of greater importance will get a smaller weight value. This has the effect of preserving more information from that image after the minimum pooling. The vector of weights W contains one value per angle, but this vector is broadcast into a matrix to enable element-wise multiplication with the input. The input to the object detector (denoted with I_m) can then be calculated by multiplying the sequence of images with the appropriate weights (W) followed by the minimum pooling along the first axis.

$$I_w = W \otimes I \tag{2a}$$

$$I_m(x,y,z) = \min_{\forall a \in \{0,1,\ldots,A-1\}} I_w(a,x,y,z) \tag{2b}$$

where:
$W \in \mathbb{R}^{A\times H\times W\times C}$ = weighting vector
$I \in \mathbb{R}^{A\times H\times W\times C}$ = input images sequence
$\otimes$ = element-wise product
A = number of used angles
H = image height
W = image width
C = number of image channels

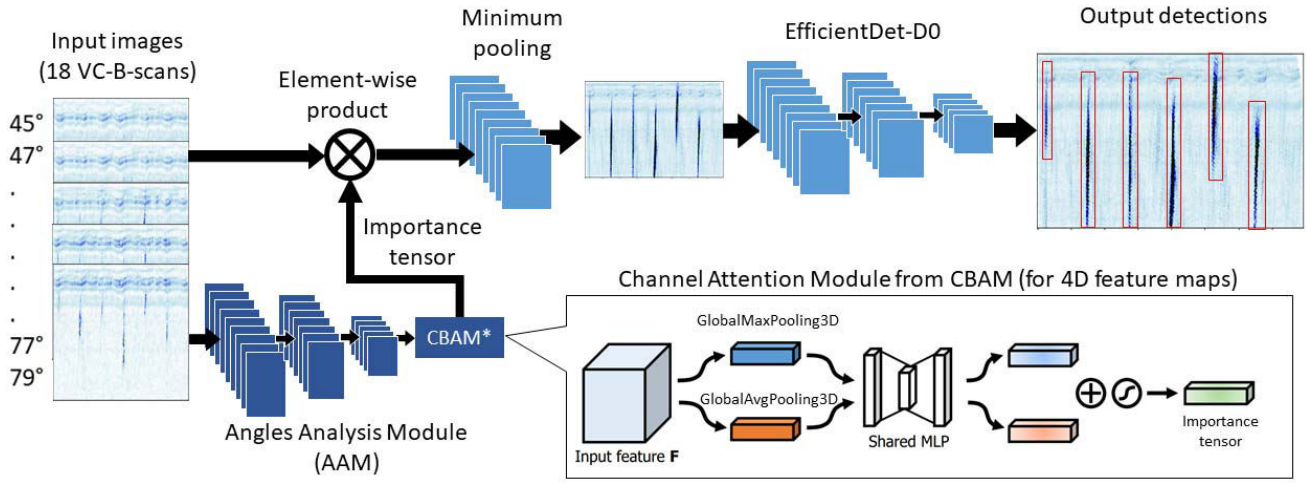


Fig. 2. Illustration of the proposed approach. Illustration for the channel attention module was taken from the original article [24].

The architecture of the AAM is shown in Table I. In-

TABLE I
ANGLE ANALYSIS MODULE ARCHITECTURE

#	Layer type	Hyperparameters	Input resolution
1	Conv3D	filters=32 kernel=(1,3,3) stride=(1,1,1)	18x384x384x3
2	MaxPooling3D	pool_size=(1,2,2)	18x384x384x32
3	Conv3D	filters=32 kernel=(1,3,3) stride=(1,1,1)	18x192x192x32
4	MaxPooling3D	pool_size=(1,2,2)	18x192x192x32
5	Conv3D	filters=32 kernel=(1,3,3) stride=(1,1,1)	18x96x96x32
6	MaxPooling3D	pool_size=(1,2,2)	18x96x96x32
7	CBAM [24]*	reduction ratio = 2	18x48x48x32

*We use only a channel attention module from CBAM.We modified the original implementation to make it compatible with our 4D input.

formation is extracted from the sequence of images using the combination of three Conv3D layers and MaxPooling3D layers. We then apply a channel attention mechanism from the convolutional bottleneck attention mechanism (CBAM) [24]. The intended usage of the original channel mechanism is to weigh the feature maps inside of some CNN. In this work, we replace the originally used GlobalAveragePooling2D and GlobalMaximumPooling2D layers with their 3D implementations in order for the module to be compatible with the dimensions of our data. The used attention mechanism feeds the extracted features into a small multi-layer perceptron (MLP) network. The hidden layer of the MLP has a lower number of neurons compared to its input and output layers. This bottleneck forces the attention mechanism to choose the important channels or in our use case the important input images. We used a reduction ratio of 2 meaning that the attention module has to pick a half of the input images that are of greater importance. Parameters of the inserted Angles Analysis Module are trained jointly with the deep learning object detection architecture. The module is independent of the used object detection architecture so it can easily be adapted to other object detectors.

III. Experimental Setup

A. Dataset

For the training and evaluation of the used models, we used an in-house dataset with over 4000 images. The dataset was obtained by scanning 6 steel blocks that contained between 6 and 34 artificially created defects inside. In total there were 68 defects. The blocks were scanned using an INETEC Dolphin scanner and a phased array probe with a frequency of 2.25 MHz. The angles between 45 and 79 with an increment of 2 degrees were used during the scanning for all of the blocks. Since the defects are artificially created inside of the blocks, their positions are known and can be used to manually annotate all of the signals that belong to the defects. Some of the defects' signals were marked even though they were barely visible. Detection of such cases is not crucial since the defects are usually seen across multiple cross-sections of the material and at different angles.

To test the possibility of simultaneous analysis of images at all angles we created a new dataset from the existing one. The dataset was created by performing the minimum merge of the images taken at different angles. For each of the resulting images, 18 input images were used. The same defects are depicted differently in each of these input images as shown in Figure 3. This illustration shows that for some of the angles the defects appear more elongated and the exact position of the defect can vary a bit. Areas of the defects' signal are increased by performing the described merging because of these variations in appearance. To label the locations of the defects in the merged images, we performed a union between

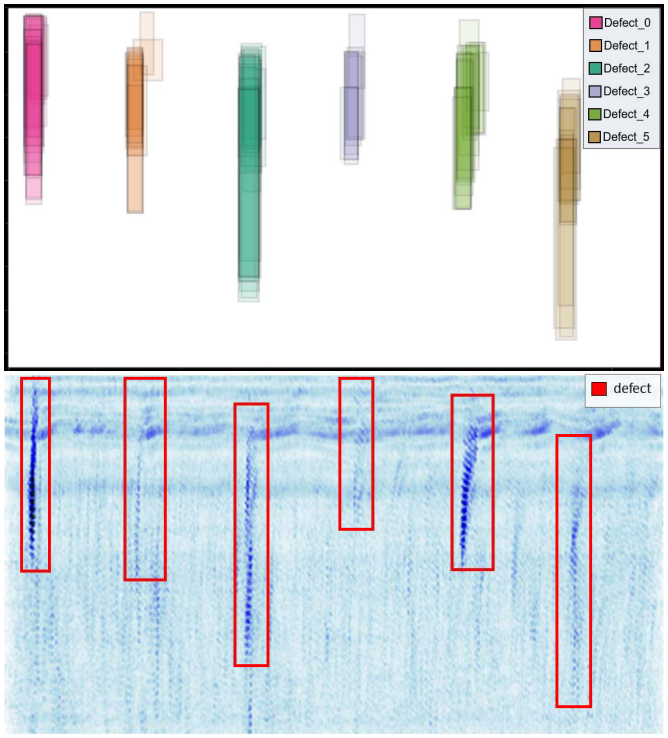


Fig. 3. Upper image displays heatmaps for all of the defects annotated for some material cross-section. It can be seen how the shape of the defects vary across different angles. Lower image shows the resulting image obtained with minimum merge of all the images displaying the same material cross-section (but at 18 different angles).

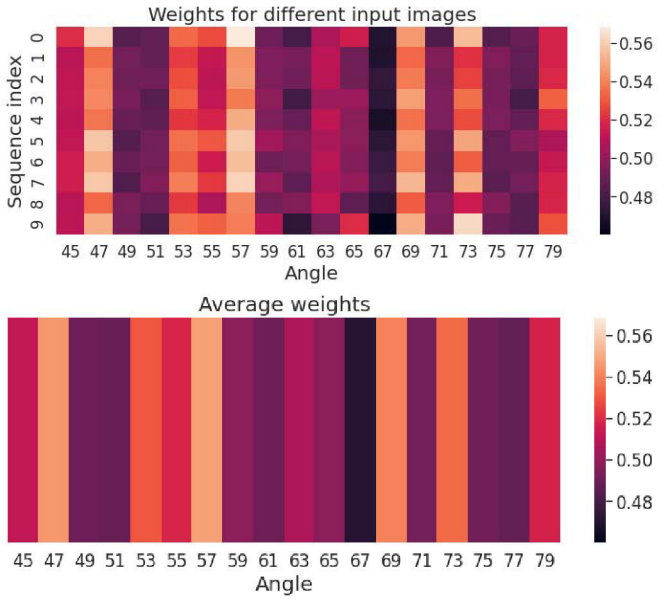


Fig. 4. The upper image displays heatmaps of weight obtained after passing random 10 samples from the test subset through the Angles Analysis Module. It can be seen how the network gives different values to each of the 18 input images. Below we showed average weights values outputted by the AAM.

all of the annotations of a defect. If the location for one of the annotations from the source images is wrong it will impact a lot the resulting annotation generated by the annotations union. We manually inspected the generated annotations for such cases and corrected them. An example of the resulting image after the merging can be seen in figure 3 down. The final dataset contained 244 images and 534 annotations. Even though image representation of UT data is naturally in the grayscale colormap, B-scans are often colored for easier manual inspection. We also used pseudo-colored images since the models were achieving equally good or better performance compared to the grayscale image analysis.

B. Experimental setup

We trained four different models in total. First, we trained the EfficientDet-D0 object detector on a full dataset. This means that the model separately analyzes images taken at different angles. We then compared three approaches that directly predict the bounding boxes of the image obtained by merging as described in section III-A. The first such model simply expands the input of the object detector from a standard 3-channel RGB image to a 54-channel input. This 54 channel input is obtained by concatenating 18 VC-B-scans taken at different angles. Next, we test a standard minimum merging. The model trained this way takes a 3-channel input image obtained by the minimum merging of the 18 input images as described in Section II. Finally, we tested the extended merging approach that dynamically determines the weights of the input images by passing them through the proposed Angles Analysis Module. The first three models from the Table II were trained with a batch size 8 and 500 steps per epoch. The last one was trained with a batch size 4 and 1000 steps per epoch. These batch sizes were the maximum batch size values that we could use (because of the memory limitation) while training these models on Nvidia Titan Xp GPU. To test the models, we randomly selected 20% of the available images. 20% of the remaining data was set aside for validation and the rest was used for training. We repeated this process 5 times and report the mean average precision (mAP), as given in the later versions of PASCAL VOC (2010-2012) [11], and a standard deviation for each of the models. During the training, we augmented the data to improve the generalization of the model and increase precision. Following transformations were used: horizontal flip, random crop, translation, and visual effects (contrast, brightness, color enhancement). The test data was not augmented.

IV. Results and Discussion

The results of the experiments are shown in Table II. The first row shows a mean average precision of the EfficientDet-D0 model trained on a full dataset. These results are not directly comparable to the rest of the results reported in the table since the angles are analyzed separately. However, this result gives a rough idea about the possible EfficientDet's performance on a defect detection task. This result also demonstrates that even when the separate analysis of images taken at various angles is performed, there will be some

TABLE II
MEAN AVERAGE PRECISION AND ANALYSIS SPEED FOR DIFFERENT APPROACHES

model	mAP	Time needed to analyze a block with 70 VC-B-scans
EfficientDet-D0 - full dataset	0.881	261 seconds
EfficientDet-D0 - input expansions (54ch)	0.843 $\sigma = 0.0148$	16.6 seconds
EfficientDet-D0 - minimum merge of input	0.850 $\sigma = 0.0146$	15.8 seconds
EfficientDet-D0 - minimum merge + AAM	**0.866** $\sigma = 0.0213$	17.8 seconds

undetected defects. We found out, by manually inspecting the false negatives, that the cases for which the model fails to detect a defect are usually some borderline cases for which the defect's signal almost completely diminished. Failing to detect these cases is not a big problem as long as the defects are detected in some other images where they are seen better. The second row of the table shows the performance of the model that simply feeds all of the images (18 images showing the same material slice but at different angles) instead of merging the images. This model is also able to inspect the data very quickly but the mean average precision is not as good as in the others tested models. The third row shows the performance of the baseline minimum merging when all of the input images are given the same importance. It can be seen that the mAP of such an approach is still a few percent lower than the expected EfficientDet's performance (first row). This is not surprising since (I) the number of images in the training subset is smaller, (II) Some of the defects' signals become hardly noticeable because of the noise introduced by the merging. The final row shows the performance of the proposed approach that extends the standard minimum merging with AAM for input image weighting. It can be seen from the results that dynamically weighting the input images before merging leads to an improvement of 1.6% compared to the uniform weighting. Even though this approach did not surpass the mAP of the model trained on a full dataset, it reaches a similar value while being a lot faster. Instead of performing separate analysis for each of the 18 angles, only one pass through the material cross-section would be needed. This leads to a speedup shown in the third column of Table II. It can be seen that the application of the proposed approach in a real-life scenario leads to a speedup of almost 15 times. In Figure 4 we showed an example of weights obtained by passing a sequence of images through AAM. The AAM gives different importance for each of the input images, and the importance of an angle is changed depending on the inputted sequence of images which is the intended behavior of the module. However, by plotting the average weights outputted by AAM, it can be concluded that some of the angles carry more information and will be given higher importance in general. The network will usually focus more on the two borderline angle values on each side (49,51 for the lower angle values and 75,77 for the larger angle values) as well as the central values of the angles (59-67).

V. CONCLUSION

In this work, we take a look at the current approach for UT data analysis and defect detection that is based on a separate analysis of VC-B-scans at each of the acquired angles. We realized that by merging the images taken at different angles we can obtain a new image that keeps the relevant information about the defects' locations. The resulting image can then be used as an input to an object detection algorithm. By performing analysis on such merged images instead of separately analyzing all of the angles, a speedup of almost 15 times is achieved in real-life scenarios. Furthermore, we proposed a novel angles analysis module that can be paired with an arbitrary deep learning object detector. We train the proposed module jointly with the object detector. The proposed module is able to determine the importance for each of the input angles and merge the images in a way that minimizes information loss and noise. We experimentally confirmed that the EfficientDet-D0 model paired with the proposed Angles Analysis Module achieves almost the same mAP as the EfficientDet-D0 model that performs the separate analysis for each of the angles. We believe that the proposed module can be further improved by using a spatial attention mechanism. This way the module could not only determine which of the input images are important but also which exact parts of the chosen images are important.

ACKNOWLEDGMENT

This research was co-funded by the European Union through the European Regional Development Fund, under the grant KK.01.2.1.01.0151 (Smart UTX). We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

REFERENCES

[1] J. Veiga, A. A. de Carvalho, I. Silva, and J. M. A. Rebello, "The use of artificial neural network in the classification of pulse-echo and tofd ultrasonic signals," *Journal of The Brazilian Society of Mechanical Sciences and Engineering - J BRAZ SOC MECH SCI ENG*, vol. 27, 10 2005. [Online]. Available: https://doi.org/10.1590/S1678-58782005000400007

[2] R. Sotero, M. Albuquerque, F. Paula, C. Farias, and E. Simas Filho, "Classification of ultrasonic signs pre-processed by fourier transform through artificial neural network using the echo pulse technique for the identification of defects in welded joints of structural steel," *Journal of Mechanics Engineering and Automation*, vol. 5, 05 2015. [Online]. Available: https://doi.org/10.17265/2159-5275/2015.05.003

[3] J. Ye, S. Ito, and N. Toyama, "Computerized ultrasonic imaging inspection: From shallow to deep learning," *Sensors*, vol. 18, no. 11, p. 3820, Nov 2018. [Online]. Available: https://doi.org/10.3390/s18113820

[4] D. Isa and R. Rajkumar, "Pipeline defect prediction using support vector machines." *Applied Artificial Intelligence*, vol. 23, pp. 758–771, 10 2009. [Online]. Available: https://doi.org/10.1080/08839510903210589

[5] A. Al-Ataby, W. Al-Nuaimy, C. Brett, and O. Zahran, "Automatic detection and classification of weld flaws in tofd data using wavelet transform and support vector machines," *Insight - Non-Destructive Testing and Condition Monitoring*, vol. 52, pp. 597–602, 11 2010. [Online]. Available: https://doi.org/10.1784/insi.2010.52.11.597

[6] Y. Wang, "Ultrasonic flaw signal classification using wavelet transform and support vector machine," *TELKOMNIKA Indonesian Journal of Electrical Engineering*, vol. 11, 07 2013. [Online]. Available: https://doi.org/10.11591/telkomnika.v11i12.3673

[7] C. Chen and G. Lee, "Neural networks for ultrasonic nde signal classification using time-frequency analysis," in *1993 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, 1993, pp. 493–496 vol.1. [Online]. Available: https://doi.org/10.1109/ICASSP.1993.319163

[8] S. Sambath, P. Nagaraj, and N. Selvakumar, "Automatic defect classification in ultrasonic ndt using artificial intelligence," *Journal of Nondestructive Evaluation*, vol. 30, no. 1, pp. 20–28, Mar 2011. [Online]. Available: https://doi.org/10.1007/s10921-010-0086-0

[9] M. Meng, Y. J. Chua, E. Wouterson, and C. P. K. Ong, "Ultrasonic signal classification and imaging system for composite materials via deep convolutional neural networks," *Neurocomputing*, vol. 257, pp. 128 – 135, 2017, machine Learning and Signal Processing for Big Multimedia Analysis. [Online]. Available: https://doi.org/10.1016/j.neucom.2016.11.066

[10] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: common objects in context," *CoRR*, vol. abs/1405.0312, 2014. [Online]. Available: https://doi.org/10.1007/978-3-319-10602-1_48

[11] M. Everingham, L. van Gool, C. Williams, J. Winn, and A. Zisserman, "The PASCAL Object Recognition Database Collection." http://host.robots.ox.ac.uk/pascal/VOC/, 2012, [Online; accessed 1-May-2020].

[12] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015. [Online]. Available: https://doi.org/10.1007/s11263-015-0816-y

[13] N. Munir, H.-J. Kim, S.-J. Song, and S.-S. Kang, "Investigation of deep neural network with drop out for ultrasonic flaw classification in weldments," *Journal of Mechanical Science and Technology*, vol. 32, no. 7, pp. 3073–3080, Jul 2018. [Online]. Available: https://doi.org/10.1007/s12206-018-0610-1

[14] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014. [Online]. Available: https://arxiv.org/abs/1409.1556

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778. [Online]. Available: 10.1109/CVPR.2016.90

[16] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," *CoRR*, vol. abs/1905.11946, 2019. [Online]. Available: http://arxiv.org/abs/1905.11946

[17] L. Posilović, D. Medak, M. Subašić, T. Petković, M. Budimir, and S. Lončarić, "Flaw detection from ultrasonic images using yolo and ssd," in *2019 11th International Symposium on Image and Signal Processing and Analysis (ISPA)*. IEEE, 2019, pp. 163–168. [Online]. Available: https://doi.org/10.1109/ISPA.2019.8868929

[18] Y. Yu, H. Cao, X. Yan, T. Wang, and S. S. Ge, "Defect identification of wind turbine blades based on defect semantic features with transfer feature extractor," *Neurocomputing*, vol. 376, pp. 1 – 9, 2020. [Online]. Available: https://doi.org/10.1016/j.neucom.2019.09.071

[19] W. Du, H. Shen, J. Fu, G. Zhang, and Q. He, "Approaches for improvement of the x-ray image defect detection of automobile casting aluminum parts based on deep learning," *NDT & E International*, vol. 107, p. 102144, 2019. [Online]. Available: https://doi.org/10.1016/j.ndteint.2019.102144

[20] X. Le, J. Mei, H. Zhang, B. Zhou, and J. Xi, "A learning-based approach for surface defect detection using small image datasets," *Neurocomputing*, vol. 408, pp. 112 – 120, 2020. [Online]. Available: https://doi.org/10.1016/j.neucom.2019.09.107

[21] K. Virupakshappa and E. Oruklu, "Automatic feature extraction based on meta-learning for ultrasonic flaw classification," in *2020 IEEE International Ultrasonics Symposium (IUS)*, 2020, pp. 1–3. [Online]. Available: https://doi.org/10.1109/IUS46767.2020.9251444

[22] R. J. Pyle, R. L. T. Bevan, R. R. Hughes, R. K. Rachev, A. A. S. Ali, and P. D. Wilcox, "Deep learning for ultrasonic crack characterization in nde," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, pp. 1–1, 2020. [Online]. Available: https://doi.org/10.1109/TUFFC.2020.3045847

[23] I. Virkkunen, T. Koskinen, O. Jessen-Juhler, and J. Rinta-aho, "Augmented ultrasonic data for machine learning," *Journal of Nondestructive Evaluation*, vol. 40, no. 1, pp. 1–11, 2021. [Online]. Available: https://doi.org/10.1007/s10921-020-00739-5

[24] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham: Springer International Publishing, 2018, pp. 3–19. [Online]. Available: https://doi.org/10.1007/978-3-030-01234-2_1

[25] M. Tan, R. Pang, and Q. V. Le, "Efficientdet: Scalable and efficient object detection," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 10 778–10 787. [Online]. Available: https://doi.org/10.1109/CVPR42600.2020.01079

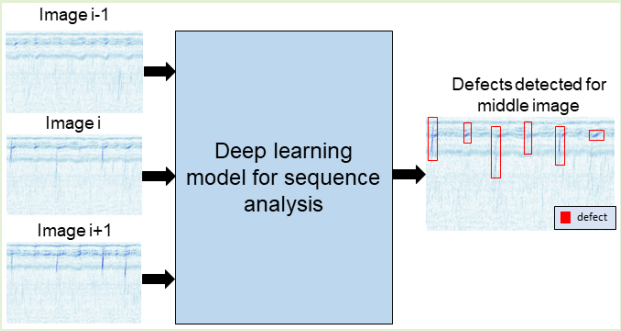
Publication 5

D. Medak, L. Posilović, M. Subašić, M. Budimir, S. Lončarić, ”Deep learning-based defect detection from sequences of ultrasonic B-scans”,*IEEE Sensors*, vol. 22, no. 3, Feb. 2022, pp. 2456-2463.

Deep Learning-Based Defect Detection From Sequences of Ultrasonic B-Scans

Duje Medak, Luka Posilović, Marko Subašić, *Member, IEEE*, Marko Budimir, *Member, IEEE*, and Sven Lončarić, *Senior Member, IEEE*

***Abstract*—Ultrasonic testing (UT) is one of the commonly used non-destructive testing (NDT) techniques for material evaluation and defect detection. The acquisition of UT data is largely performed automatically by using various robotic manipulators which can ensure the consistency of the recorded data. On the other hand, complete analysis of the acquired data is still performed manually by trained personnel. This makes the reliability of defect detection highly dependent on humans' knowledge and experience. Most of the previous attempts for automated defect detection from UT data analyze individual A-scans. In such cases, valuable information present in the surrounding A-scans remains unused and limits the performance of such methods. The situation is better if a B-scan is used as an input, especially if the dataset is large enough to train a deep learning object detector. However, if each of the B-scans is analyzed individually, as it was done so far in the literature, there is still valuable information left in the surrounding B-scans that could be used to improve the precision. We showed that expanding the input layer of an existing method will not lead to an improvement and that a more complex approach is needed in order to effectively use information from neighboring B-scans. We propose two approaches based on high-dimensional feature maps merging. We showed that proposed models improve mean average precision (mAP) compared to the previous state-of-the-art model by 2% for input resolutions of 512 × 512 pixels, and 3.4% for input resolutions of 384 × 384 pixels.**



***Index Terms*—Image analysis, deep learning, convolutional neural networks, defect detection, ultrasonic testing.**

I. Introduction

NON-DESTRUCTIVE testing (NDT) is a group of techniques for evaluation of material's properties and flaw detection, commonly used in industry and science [1]. NDT includes a variety of methods such as eddy current, thermography, radiography, and ultrasonic testing. Being non-destructive by its nature makes the mentioned approaches popular for continuous monitoring of critical components of some systems. Some of the areas where NDT is often used include the oil and gas industry, various power plants, and aeronautics. Ultrasonic testing (UT) has several advantages compared to other NDT methods. It is simple to employ, enables precise extraction of the defect location [2], and has a high signal-to-noise ratio [3]. UT can be implemented using different technologies but the main idea is based upon the generation and detection of mechanical vibrations or waves within test objects [4]. Ultrasonic waves can be produced and received by the same probe (ultrasound transducer). One of the commonly used types of probes is called a phased array. A phased array probe is a multi-channel ultrasonic system, which uses the principle of a time-delayed triggering of the transmitting transducer elements, combined with a time corrected receiving of detected signals [5]. This probe enables simultaneous inspection of the material from different angles which increases the reliability of flaw detection. During an inspection, a robotic manipulator moves the ultrasonic probe along the surface of the inspected component. At each position, the probe transmits ultrasonic waves and receives the reflected signals. The value of the reflected signal is altered if the wave is reflected from an object with a different density compared to the surrounding area. This property makes the detection of various types of flaws possible. The received signal can be displayed in different forms as shown in Figure 1. During the analysis, trained experts manually inspect the acquired data to localize

Manuscript received October 13, 2021; revised December 7, 2021; accepted December 7, 2021. Date of publication December 9, 2021; date of current version January 31, 2022. This work was supported by the European Union through the European Regional Development Fund [smart modular system for ultrasound diagnostics in extreme conditions (Smart UTX)] under Grant KK.01.2.1.01.0151. The associate editor coordinating the review of this article and approving it for publication was Dr. Varun Bajaj. *(Duje Medak and Luka Posilović contributed equally to this work.) (Corresponding author: Duje Medak.)*

Duje Medak, Luka Posilović, Marko Subašić, and Sven Lončarić are with the Faculty of Electrical Engineering and Computing, University of Zagreb, 10000 Zagreb, Croatia (e-mail: duje.medak@fer.hr).

Marko Budimir is with Inetec Ltd., 10250 Zagreb, Croatia.

Digital Object Identifier 10.1109/JSEN.2021.3134452

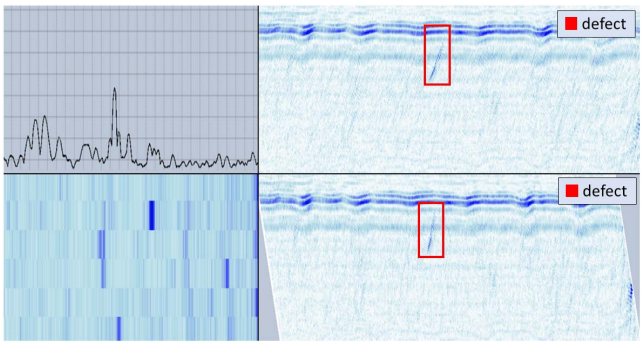


Fig. 1. Examples of different ultrasonic data representations [10]. Upper left image shows an A-scan. Beneath it is the C-scan representation. Right side of the image displays B-scan (top) and volume corrected B-scan (bottom).

and size the defects. The amount of data that needs to be analyzed in the majority of the cases is immense, especially if a phased array system is used. Even though the inspectors go through years of training, the repetitive work of UT data analysis can lead to fatigue and some of the flaws might go unnoticed due to human error, not to mention the considerable amount of time and personnel needed for such activity. These are the key motivations for the development of a method that could non-trivially assist the inspectors with the analysis of the data, or even automate this whole process in the future. An overview of existing methods for the analysis of NDT data is given in Section II. The main problem with the development of a reliable method for defect detection is that the work logic of the human inspectors can not be simply expressed with an algorithmic description [6]. A more complex approach based on machine learning or deep learning is needed. It was already shown in the literature [2], [7]–[9] that deep learning models outperform methods based on hand-crafted features and classical image analysis algorithms. However, training a deep learning model requires a large dataset of labeled data which can be difficult to obtain.

Defects in UT data usually span across multiple B-scans. The signal of a defect can be very weak, especially on the first and the last B-scan displaying the defect so it is very difficult to distinguish such signal from the noise. In real-life scenarios, inspectors would take a look at the surrounding area or the same coordinates from a B-scan viewed from a different angle to confirm their decision. However, currently proposed solutions for automated defect detection do not use this additional context to improve the precision of classification and localization. In previous work [11] it was indicated that the lack of context from the neighboring slices is the main limitation of defect detector's precision. To the best of our knowledge, a method that performs defect detection by looking at the multiple B-scan images (surrounding volume) was not yet proposed in the literature. In this work, we propose several methods that can be used to include information from the neighboring B-scans. We first showed an important fact that the simple expansion of the input does not improve detector's mean average precision (mAP). This was tested by expanding the input layer of EfficientDet-D0 architecture, a model that was already proved to work well for the defect detection task [11]. Feeding a 9 channel input, produced by concatenation of three neighboring B-scans, did not increase the precision of the model demonstrating the need for a more complex solution. We then propose two other novel approaches that successfully use the additional information by performing high-dimensional feature maps merging. Proposed approaches first separately extract multi-scale features from the sequence of B-scans and then combine all of the extracted features before localizing the defects. We demonstrated that the proposed architectures significantly improve mean average precision compared to the baseline 2D model.

II. Related Work

Since the results of the NDT data analysis mostly depend on the abilities of the inspectors, they are susceptible to human errors and can thus be inconsistent and unreliable. To tackle this problem and assist inspectors with the analysis, many methods for signal and image processing methods were developed. Those methods can work with different types of NDT data such as the data acquired during a visual inspection [7], [8], thermography inspection [12]–[14], radiography inspection [15], [16], or ultrasonic inspection [6], [9], [17]–[20]. Since the format of the acquired data depends on the used technique, the exact implementation of these methods differs. However, the main idea of the proposed methods are usually similar and rely either on hand-crafted descriptors in combination with some classifier or the direct application of convolutional neural networks (CNNs). Because of their structure that is based on a series of convolution operations, CNNs can naturally process sequences and grid-like representations of the data. CNNs usually outperform classical approaches [2], [7]–[9] so it is not surprising that this approach became the most popular choice in recent years. [6]–[9], [12]–[18], [20].

Before CNNs began to be used for the analysis of UT data, the most popular approach for defect detection was based on the analysis of ultrasonic A-scans using the wavelet transform in combination with some classifiers. Commonly used classifiers were Artificial Neural Networks (ANN), used in [21], [22], and Support Vector Machine (SVM) used in [23]–[26]. The main drawback of A-scan analysis is the lack of valuable information from the surrounding area of some A-scan. Having this information would make it easier to distinguish between defect signal and geometry or noise signal. This is why the analysis of ultrasonic B-scans is becoming more popular nowadays, but the unavailability of the image data is a major factor limiting the research of the automated analysis of UT images. This can partially be solved by using transfer learning [27] and data augmentation techniques [28]. In [18] the authors applied those techniques and trained existing one-stage called detectors Single Shot Detector (SSD) and You Only Look Once (YOLO) to detect defects from ultrasonic B-scan. The reported results were promising but the dataset used for development and testing contained only several hundred images. In [17] the authors used ultrasonic images simulated with OnScale software to create their dataset. They simulated 400 B-scans with four different types of defects. The authors evaluated several different CNN architectures and the best among the tested ones achieved an average accuracy greater than 92%. In [29], the authors trained a VGG-like [30] CNN

TABLE I
DATASET OVERVIEW

	number of defects	number of images	number of annotations
fold 1	14	1006	1317
fold 2	15	915	1439
fold 3	16	872	1437
fold 4	12	298	1316
fold 5	11	1083	1128
total	68	4174	6637

on artificially generated ultrasonic images. The performance of the developed model was compared to the performance of a human expert. The CNN model detected two cracks less compared to the human inspector but the inspector made more false calls. The authors also noted the importance of training data selection on the model's performance. In [11], several one-stage object detectors were tested on the largest dataset of real UT B-scans that was by that time used in the literature. The best results among the tested models were achieved by the EfficientDet-D0 architecture that reached a mean average precision of 89.6%. The authors showed that even though a perfect precision of 100% was not achieved, all of the defects were detected because each defect appears on several B-scans and not all of the appearances need to be detected. It was pointed out that false detections are usually caused by a lack of context. This means that one of the possible ways to improve the precision is taking into account the surrounding area of inputted B-scan which is done in this work. In [31] the authors demonstrated that a CNN mostly trained with simulated data and with a small amount of experimental data can be used to detect, locate and size a defect from ultrasonic phased array data. The used dataset was created by GPU-accelerated finite element simulations and then expanded with a small percentage of real data. The authors trained a two-stage detector Faster-RCNN that reached the area under the curve of 0.95 when tested on simulated data. When testing the detector on real data with an intersection over union (IOU) threshold of 0.4, the model was able to locate 70% of the flat bottom hole defects. All of the aforementioned methods confirm that the most promising approach at the moment is to use a deep convolutional neural network to analyze ultrasonic images and detect defects.

III. DATASET

For the training and evaluation of the proposed method, we used an in-house dataset. The data was obtained by scanning 6 stainless steel blocks for internal UT acquisition equipment qualification training. The blocks contained between 6 and 34 defects. Defects were artificially created using various methodologies leading to different types of defects: side-drilled holes, flat bottom holes, thermal fatigue cracks, mechanical fatigue cracks, electric discharge machined notches, solidification cracks, and incomplete penetration of the weld. The scanning was done by the INETEC Dolphin pulser-receiver instrument and a phased array probe with a central frequency of 2.25 MHz. The collected data used for this work includes only the shallower parts of the blocks (down to 200mm depth). After all the needed data was collected, multiple human experts analyzed it and determined the positions of the defects. The location of each defect was annotated by a bounding box. Defects in B-scans usually appear slanted so the bounding box would not fit perfectly around the defect. Therefore we have decided to use volume corrected B-scans (VC-B-scans). In VC-B-scans each A-scan is transferred onto the image at the same angle that the ultrasonic waves were propagated through the material. This skews VC-B-scans as shown in Figure 1 and keeps the orientations of the displayed defects more similar to the physical orientation of the defects inside of the material. Defects in the VC-B-scan appear vertical so the exact location is more precisely captured with a rectangle bounding box. After gathering all of the images and annotations we split the data into 5 subsets/folds. Each fold contains unique defects as seen in Table I. The dataset is the same as in [11] and the results of the best performing model EfficientDet-D0 were taken as baseline results in this work. The procedure for splitting was designed in a way that allows expansion of the images with the neighboring slices without giving an unwanted advantage to the model trained this way. In other words, surrounding images for each of the images in the training subsets are also always contained in the training subset. This ensures that all of the sequences used for testing as well as the defects that are displayed in those sequences are unique and will not be seen during the training. The sequences we used in this work were created using only the immediate neighbors of some image so all of the sequences consist of three images. This is the smallest odd number of B-scans that can be used for defect detection that includes some additional context. We believe that looking at the closest neighboring images is enough to decide whether the target (middle) B-scan contains a defect at some specific location or not. We did not perform experiments with the sequences of greater length since that would decrease the number of samples in our dataset and also increase the time needed for a forward pass through the model. However, the approaches we propose in this work are not in any way limited to sequences of length three and can easily be extended to work with an arbitrary number of input B-scans. For the target images that only have one neighboring slice, we replace the missing slice with the existing neighboring B-scan. All of the images from one sequence have the same height and width. The height of the images varies between 200 and 375 pixels while their width varies between 300 and 400 pixels. A few example images from the dataset are shown in Figure 2.

IV. METHODOLOGY

A. Baseline Architecture - EfficientDet-D0

EfficientDet [32] is a family of object detection models proposed by the Google research team in 2019. The authors proposed a novel baseline architecture called EfficientDet-D0 that can be scaled up depending on the available resources. This idea of scaling a baseline model was already proven to work well with the EfficientNet [33] classification architecture. This model was thus a logical feature extractor choice since it can easily be scaled up alongside the other parts of the object detection model. The scaling is performed jointly for all of the components of the detection model (backbone, feature

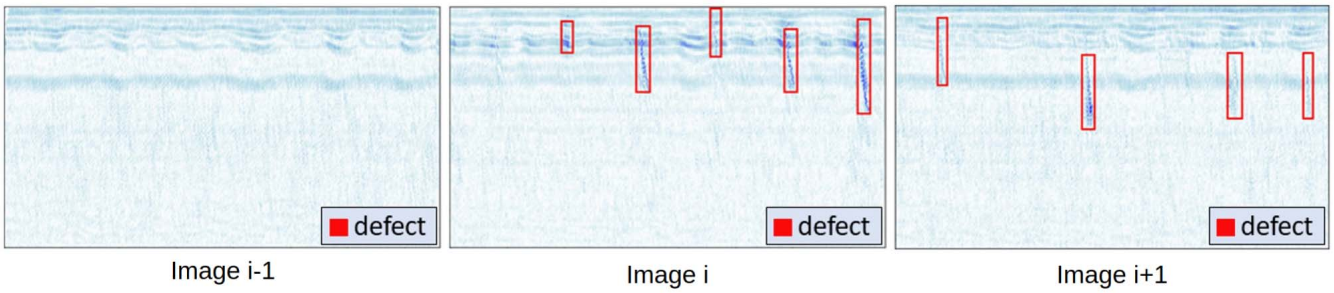


Fig. 2. An example sequence of VC-B-scans with ground truth labels.

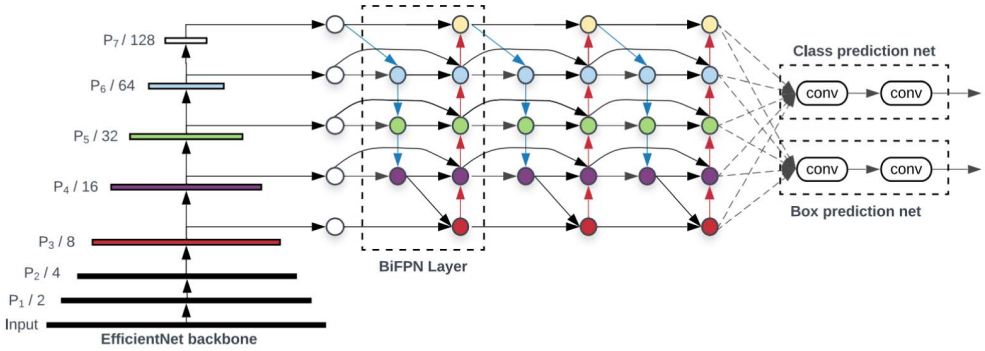


Fig. 3. Illustration of the EfficientDet architecture. source [32].

network, and detection head). The authors used a heuristic-based approach for determining the best scaling coefficient values. The EfficientDet family of object detectors consists of eight object detectors in total. Several of those detectors were tested for a defect detection task in [11] and it was shown that the smallest of the tested architectures achieved the best mean average precision. This is probably caused by the simplicity of the ultrasonic dataset compared to the large-scale dataset such as ImageNet [34] and COCO [35] that are commonly used for the development of object detectors. Also, since our dataset is quite small, a model with fewer parameters can be trained more easily. However, the methods that we use to expand the analysis from one image to the analysis of sequences of images are not dependant on the choice of object detector and so they can be applied to other models as well.

B. Approaches for Sequence Analysis

An arbitrary object detection model that was developed for image analysis can be expanded to work with a sequence of images in several ways:

(I) The simplest approach for including the surrounding images while performing the object detection is to expand the input dimensions of the model. In this case, one training example would have dimensions: (image height, image width, N $\times$ number of channels), where N is the number of images in the sequence. In this work, we set the value of N to three (only immediate neighbors are considered). Information from all of the VC-B-scans is simultaneously extracted while passing through the network. If the number of filters is not increased compared to the baseline network, the computational overhead of the described modification is negligible.

(II) Another approach is to separately extract the features from all of the images in the sequence and merge the obtained features before the detection head. We decided to perform merging after the feature network (BiFPN). We experimentally determined that merging the feature maps at this stage works better than merging the feature maps before the feature network. The merging can also be performed in other stages of the network but the two mentioned and tested positions are the most logical choices. As shown in Figure 3, the feature network outputs feature maps at 5 different scales. Dimensions of these feature maps are equivalent to the ones outputted by the feature extractor. For a standard EfficientDet-D0 model, mentioned feature maps have a depth of 64 and spatial resolution spanning from 4×4 for the P7 feature map, up to 64×64 for the P3 feature map. If we separately pass the sequence of N images through the backbone, we will obtain N feature maps for each of the levels (P3-P7). We can then fuse the information from multiple feature maps by using the standard convolutional layer or by using a convolutional long short-term memory (ConvLSTM) layer. Illustrations of these approaches are shown in Figures 4 and 5.

If the standard (two-dimensional) convolutional layers are used to merge features, the outputs of the feature extractor first need to be concatenated along the channel axis. The resulting feature maps will then have 192 channels, regardless of the spatial resolution. To force the model to choose important features from the sequences of images, we inserted a convolutional layer that decreases the number of channels by three times. The resulting layers contain information extracted from the whole sequence and their shapes are the same as the shapes of the original feature extractor (and feature network) outputs. This means that the original detection head from EfficientDet

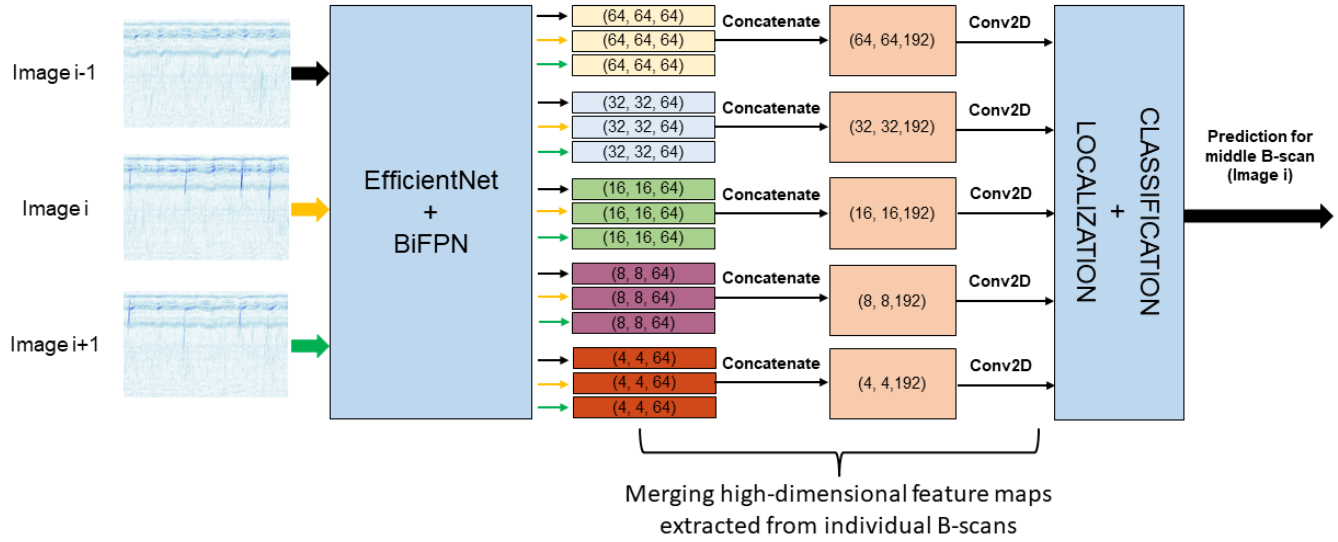


Fig. 4. Merging feature maps from neighboring slices using the Conv2D layer.

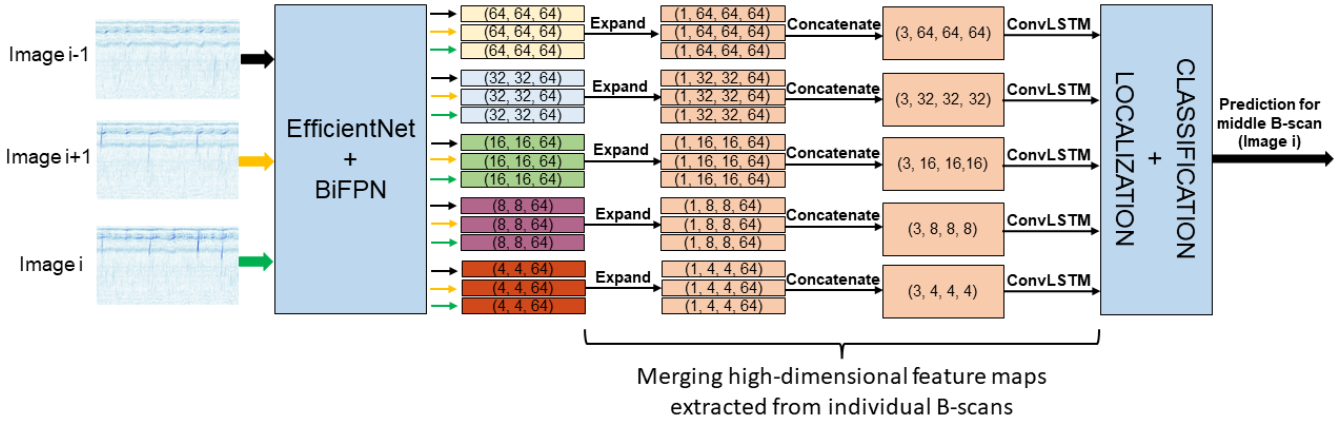


Fig. 5. Merging feature maps from neighboring slices using the ConvLSTM layer.

architecture can be used without any modifications. We tested several values for kernel size and activation function of the inserted layer. We experimentally determined that the best combination is to simply use a 1×1 convolution without activation function. The described process of feature maps merging can be described as written in Equation 1.

$$P_p^{out} = \text{Conv}\left(\text{Concatenate}(P_p^{i-1}, P_p^i, P_p^{i+1})\right) \quad (1)$$

where:

P_p^i	=feature map of level p obtained by passing image i through the feature extractor
$p \in \{3, 4, 5, 6, 7\}$	=pyramid level
i	=index of the image that is being analyzed
Concatenate	=concatenation operation along the last (channel) axis

Another approach for fusing feature maps from the sequence of images is by using a ConvLSTM layer [36]. Long Short Term Memory (LSTM) is a type of Recurrent Neural Network that is useful for the analysis of data collected over time. This type of model has an internal state in which the most important information about the previously seen inputs can be stored. Having a glimpse at previously inputted data can be very useful when deciding on the current input. In our case, the observed time series is a sequence of images. Since convolutional layers are commonly used to build models for

TABLE II
BATCH SIZE AND NUMBER OF STEPS FOR EACH OF THE TESTED MODELS

model	batch size	steps per epoch
EfficinetDet-D0-512-ConvLSTM EfficinetDet-D0-512-Conv2D	2	2000
EfficinetDet-D0-384-ConvLSTM EfficinetDet-D0-384-Conv2D	4	1000
EfficinetDet-D0-512 EfficinetDet-D0-512-9ch EfficinetDet-D0-384 EfficinetDet-D0-384-9ch	8	500

TABLE III
MEAN AVERAGE PRECISION (MAP) AND INFERENCE TIME FOR MODELS WITH INPUT RESOLUTION OF $3 \times 512 \times 512 \times 3$. BOLD TEXT INDICATES THE BEST PERFORMANCE FOR THAT FOLD

model	fold1	fold2	fold3	fold4	fold5	average	inference time
EfficientDet-D0	0.937	0.829	0.879	**0.943**	0.893	0.896	42.0 ms
EfficientDet-D0 - 9 channel input	0.913	0.817	0.877	0.934	0.939	0.896	39.3 ms
EfficientDet + ConvLSTM merging	0.919	**0.888**	**0.897**	0.913	0.920	0.907	84.8 ms
EfficientDet-D0 + Conv2D merging	**0.942**	0.874	0.896	0.938	**0.933**	**0.916**	66.1 ms

image analysis it would be beneficial to use convolution operation in combination with LSTM to analyze sequences of images. This is why in this work we use a special type of LSTM called ConvLSTM [36]. ConvLSTM layer replaces the internal matrix multiplication of a standard LSTM with the convolution operation. This modification helps the network learn dependencies along the time (depth) axis while preserving the spatial information in the feature maps.

$$\begin{aligned} PE_p^{i-1} &= \text{Expand}\left(P_p^{i-1}\right) \\ PE_p^{i+1} &= \text{Expand}\left(P_p^{i+1}\right) \\ PE_p^{i} &= \text{Expand}\left(P_p^{i}\right) \\ P_p^{out} &= \text{ConvLSTM}\left(\text{Concatenate}(PE_p^{i-1}, PE_p^{i+1}, PE_p^{i})\right) \end{aligned} \tag{2}$$

where:

P_p^i	=feature map of level p obtained by passing image i through the feature extractor
$p \in \{3, 4, 5, 6, 7\}$	=pyramid level
i	=index of the image that is being analyzed
$Concatenate$	=concatenation operation along the first (expanded) axis

This layer requires the data with an extra dimension that represents the temporal axis. We first calculate the feature maps for each of the images in the sequence. We then expand the dimensions of these feature maps by inserting another axis at the beginning of the tensor. Obtained feature maps can now be concatenated along the first (inserted) axis and used as an input to ConvLSTM. The most recent feature map that is inputted into the ConvLSTM layer is the one for which the prediction is made so it is important to reshape the input tensor correctly. We do this intentionally so that the most important features that are calculated from the middle image are preserved the best. The described process of feature maps merging with ConvLSTM can be written down as shown in Equation 2. ConvLSTM layer can return a whole sequence as an output but since we do not perform additional analysis after merging the feature this option was not used. The dimensions of the feature maps calculated using the ConvLSTM are the same as the ones calculated with Conv2D from the previous approach. This means that the detection head of the standard EfficientDet-D0 does not need to be modified.

V. EXPERIMENTAL SETUP AND RESULTS

A. Experimental Setup

Proposed methods were tested on the dataset described in Section III. We used mean average precision averaged across five folds to evaluate the performance of the models. We used mAP calculation as given in the later versions of PASCAL VOC (2010-2012) [37], with an intersection over union (IOU) threshold of 0.5. We trained and evaluated all the models' variations with two spatial input resolutions: 512×512 and 384×384. Pretrained weights obtained by training the models on the COCO dataset were used as a starting point in each of the experiments. All models were trained using the ADAM optimizer with an initial learning rate of $1e^{-3}$. The learning rate was reduced by 10 times if there were no improvements of mean average precision on the validation subset for six consecutive epochs. The training of the models was stopped if there were no improvements of mean average precision on the validation subset for 10 consecutive epochs. The batch size and number of steps for each of the models can be seen in Table II. The training subset was augmented during the training which is commonly done to improve the generalization of the model and increase precision. Following transformations were used: horizontal flip, random crop, translation, and visual effects (contrast, brightness, color enhancement). When training the models on sequence data, all the images in the sequence were augmented using the same transformations.

B. Results and Discussion

The results of the evaluation can be seen in Tables III and IV. Expanding the input of the model to simultaneously analyze all of the images from the sequence does not lead to an improvement for the bigger (512×512) model. For the smaller model, the improvement of such an approach is very small. However, the other two presented approaches for sequence analysis increase the mean average precision by 2% and 3.4% for the models with input resolutions of 512×512 and 384×384 respectively. The approach based on Conv2D merging worked better for the larger input resolution while the approach based on ConvLSTM worked better for the smaller one. These results show the effectiveness of the proposed approaches and prove how useful the information from the surrounding VC-B-scans can be when performing defect detection. The increased precision comes at the cost of the higher inference speed. The method for merging the feature maps using the convolutional layer takes 60% longer compared to the baseline method with an input resolution of 512×512. The method based on the

TABLE IV
MEAN AVERAGE PRECISION (MAP) AND INFERENCE TIME FOR MODELS WITH INPUT RESOLUTION OF $3 \times 384 \times 384 \times 3$. BOLD TEXT INDICATES THE BEST PERFORMANCE FOR THAT FOLD

model	fold1	fold2	fold3	fold4	fold5	average	inference time
EfficientDet-D0	0.903	0.816	0.882	0.926	0.878	0.881	35.6 ms
EfficientDet-D0 - 9 channel input	0.895	0.815	0.857	0.939	**0.929**	0.887	32.6 ms
EfficientDet + ConvLSTM merging	**0.927**	**0.895**	**0.890**	0.934	0.921	**0.914**	80.3 ms
EfficientDet-D0 + Conv2D merging	0.917	0.869	0.874	**0.944**	0.921	0.905	63.2 ms

ConvLSTM layer needs twice as much time as the baseline model for that same resolution. However, the inference time in a real-life scenario could be significantly decreased. If the long sequence of VC-B-scans needs to be analyzed, it is possible to reuse the feature maps that were already calculated. This way each image would pass through the feature extractor only once and the only additional computational complexity would come from the feature merging which would not significantly slow down the model. We also wanted to note that the time reported for the model with a nine-channel input is not a mistake. That model should theoretically take longer for inference but in our experiments, we got a slightly faster time compared to the baseline model. We believe that this anomaly is caused by the better optimization of that model for GPU execution.

VI. CONCLUSION

In this paper, we propose two novel approaches that can be used to incorporate information from the surrounding B-scans while performing defect detection from ultrasound images. We tested the effectiveness of the proposed approaches on the in-house dataset with over 4000 sequences of VC-B-scans. We showed that the simple expansion of the EfficientDet's input does not lead to a significant improvement proving the need for a more sophisticated approach for sequence analysis. Using the methods proposed in this work, the mean average precision can be improved by 2.0 % for the model with an input resolution of 512×512 and by 3.4% for the model with an input resolution of 384×384. In the future, some other approaches for feature merging as well as some other object detection models should be considered.

REFERENCES

[1] L. Cartz, *Nondestructive Testing: Radiography, Ultrasonics, Liquid Penetrant, Magnetic Particle, Eddy Current*. Materials Park, OH, USA: ASM Int., 1995. [Online]. Available: https://books.google.hr/books?id=0spRAAAAMAAJ

[2] J. Ye, S. Ito, and N. Toyama, "Computerized ultrasonic imaging inspection: From shallow to deep learning," *Sensors*, vol. 18, no. 11, p. 3820, 2018, doi: 10.3390/s18113820.

[3] P. Broberg, "Imaging and analysis methods for automated weld inspection," Ph.D. dissertation, Dept. Eng. Sci. Math., Division Fluid Exp. Mech., Luleå Tekniska Universitet, Luleå, Sweden, 2014.

[4] D. Forsyth, "Nondestructive testing of corrosion in the aerospace industry," in *Corrosion Control in the Aerospace Industry* (Woodhead Publishing Series in Metals and Surface Engineering), S. Benavides, Ed. Sawston, U.K.: Woodhead Publishing, 2009, ch. 5, pp. 111–130. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9781845693459500050

[5] L. von Bernus, A. Bulavinov, D. Joneit, M. Kröning, M. Dalichov, and K. M. Reddy, "Sampling phased array: A new technique for signal processing and ultrasonic imaging," in *Proc. Eur. Conf. NonDestructive Test. (ECNDT)*, Berlin, Germany, 2006. [Online]. Available: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.218.3412&rep=rep1&type=pdf

[6] I. Virkkunen, T. Koskinen, O. Jessen-Juhler, and J. Rinta-aho, "Augmented ultrasonic data for machine learning," *J. Nondestruct. Eval.*, vol. 40, no. 1, pp. 1–11, Mar. 2021.

[7] J. Masci, U. Meier, D. Ciresan, J. Schmidhuber, and G. Fricout, "Steel defect classification with max-pooling convolutional neural networks," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jun. 2012, pp. 1–6.

[8] Y. Yu, H. Cao, X. Yan, T. Wang, and S. S. Ge, "Defect identification of wind turbine blades based on defect semantic features with transfer feature extractor," *Neurocomputing*, vol. 376, pp. 1–9, Feb. 2020. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925231219313396

[9] M. Meng, Y. J. Chua, E. Wouterson, and C. P. K. Ong, "Ultrasonic signal classification and imaging system for composite materials via deep convolutional neural networks," *Neurocomputing*, vol. 257, pp. 128–135, Dec. 2017, doi: 10.1016/j.neucom.2016.11.066.

[10] B. Sheeba and P. Myvizhi, "Industrial applications of A, B and C scan mode," *Int. J. Pure Appl. Math.*, vol. 119, no. 12, pp. 3861–3872, 2018.

[11] D. Medak, L. Posilovic, M. Subasic, M. Budimir, and S. Loncaric, "Automated defect detection from ultrasonic images using deep learning," *IEEE Trans. Ultrason., Ferroelectr., Freq. Control*, vol. 68, no. 10, pp. 3126–3134, Oct. 2021.

[12] Q. Luo, B. Gao, W. L. Woo, and Y. Yang, "Temporal and spatial deep learning network for infrared thermal defect detection," *NDT E Int.*, vol. 108, Dec. 2019, Art. no. 102164. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0963869519301355

[13] L. Ruan, B. Gao, S. Wu, and W. L. Woo, "DeftectNet: Joint loss structured deep adversarial network for thermography defect detecting system," *Neurocomputing*, vol. 417, pp. 441–457, Dec. 2020. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925231220312637

[14] H.-T. Bang, S. Park, and H. Jeon, "Defect identification in composite materials via thermography and deep learning techniques," *Compos. Struct.*, vol. 246, Aug. 2020, Art. no. 112405. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S026382232030146X

[15] W. Du, H. Shen, J. Fu, G. Zhang, and Q. He, "Approaches for improvement of the X-ray image defect detection of automobile casting aluminum parts based on deep learning," *NDT E Int.*, vol. 107, Oct. 2019, Art. no. 102144. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0963869519300192

[16] X. Le, J. Mei, H. Zhang, B. Zhou, and J. Xi, "A learning-based approach for surface defect detection using small image datasets," *Neurocomputing*, vol. 408, pp. 112–120, Sep. 2020. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925231220303386

[17] K. Virupakshappa and E. Oruklu, "Multi-class classification of defect types in ultrasonic NDT signals with convolutional neural networks," in *Proc. IEEE Int. Ultrason. Symp. (IUS)*, Oct. 2019, pp. 1647–1650.

[18] L. Posilovic, D. Medak, M. Subasic, T. Petkovic, M. Budimir, and S. Loncaric, "Flaw detection from ultrasonic images using Yolo and SSD," in *Proc. 11th Int. Symp. Image Signal Process. Anal. (ISPA)*, Sep. 2019, pp. 163–168.

[19] N. Munir, H.-J. Kim, S.-J. Song, and S.-S. Kang, "Investigation of deep neural network with drop out for ultrasonic flaw classification in weldments," *J. Mech. Sci. Technol.*, vol. 32, pp. 3073–3080, 2018, doi: 10.1007/s12206-018-0610-1.

[20] N. Munir, H.-J. Kim, J. Park, S.-J. Song, and S.-S. Kang, "Convolutional neural network for ultrasonic weldment flaw classification in noisy conditions," *Ultrasonics*, vol. 94, pp. 74–81, Apr. 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0041624X18305754

[21] F. Bettayeb, T. Rachedi, and H. Benbartaoui, "An improved automated ultrasonic NDE system by wavelet and neuron networks," *Ultrasonics*, vol. 42, no. 1, pp. 853–858, 2004, doi: 10.1016/j.ultras.2004.01.064.

[22] S. Sambath, P. Nagaraj, and N. Selvakumar, "Automatic defect classification in ultrasonic NDT using artificial intelligence," *J. Nondestruct. Eval.*, vol. 30, pp. 20–28, Mar. 2011, doi: 10.1007/s10921-010-0086-0.

[23] M. Khelil, M. Boudraa, A. Kechida, and R. Drai, "Classification of defects by the SVM method and the principal component analysis (PCA)," *Int. J. Electr. Comput. Eng.*, vol. 1, no. 9, pp. 1–6, 2007, doi: 10.5281/zenodo.1060751.
[24] V. Matz, M. Kreidl, and R. Smid, "Classification of ultrasonic signals," *Int. J. Mater. Product Technol.*, vol. 27, nos. 3–4, p. 145, 2006, doi: 10.1504/IJMPT.2006.011267.
[25] A. Al-Ataby, W. Al-Nuaimy, C. R. Brett, and O. Zahran, "Automatic detection and classification of weld flaws in TOFD data using wavelet transform and support vector machines," *Insight-Non-Destruct. Test. Condition Monit.*, vol. 52, no. 11, pp. 597–602, Nov. 2010, doi: 10.1784/insi.2010.52.11.597.
[26] Y. Chen, H.-W. Ma, and G.-M. Zhang, "A support vector machine approach for classification of welding defects from ultrasonic signals," *Nondestruct. Test. Eval.*, vol. 29, no. 3, pp. 243–254, Jul. 2014, doi: 10.1080/10589759.2014.914210.
[27] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features Off-the-shelf: An astounding baseline for recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2014, pp. 512–519.
[28] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *J. Big Data*, vol. 6, p. 60, Jul. 2019, doi: 10.1186/s40537-019-0197-0.
[29] O. Siljama, T. Koskinen, O. Jessen-Juhler, and I. Virkkunen, "Automated flaw detection in multi-channel phased array ultrasonic data using machine learning," *J. Nondestruct. Eval.*, vol. 40, no. 3, Aug. 2021.
[30] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.
[31] T. Latête, B. Gauthier, and P. Belanger, "Towards using convolutional neural network to locate, identify and size defects in phased array ultrasonic testing," *Ultrasonics*, vol. 115, Aug. 2021, Art. no. 106436. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0041624X21000731
[32] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and efficient object detection," 2019, *arXiv:1911.09070*.
[33] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," 2019, *arXiv:1905.11946*.
[34] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.
[35] T. Y. Lin *et al.*, "Microsoft COCO: Common objects in context," in *Computer Vision—ECCV 2014* (Lecture Notes in Computer Science), vol. 8693, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham, Switzerland: Springer, 2014, doi: 10.1007/978-3-319-10602-1_48.
[36] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-C. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Proc. 28th Int. Conf. Neural Inf. Process. Syst.*, vol. 1. Cambridge, MA, USA: MIT Press, 2015, pp. 802–810.
[37] M. Everingham, L. van Gool, C. Williams, J. Winn, and A. Zisserman. (2012). *The PASCAL Object Recognition Database Collection*. Accessed: May 1, 2020. [Online]. Available: http://host.robots.ox.ac.uk/pascal/VOC/

Duje Medak received the M.Sc. degree from the Faculty of Electrical Engineering and Computing, University of Zagreb, in 2019, where he is currently pursuing the Ph.D. degree with the Faculty of Electrical Engineering and Computing. He is working as a Researcher with the Image Processing Group, Department of Electronic Systems and Information Processing. His research interests include image processing, image analysis, machine learning, and deep learning. His current research interests include deep learning object detection methods and their application in the nondestructive testing (NDT) domain.

Luka Posilović received the M.Sc. degree from the Faculty of Electrical Engineering and Computing, University of Zagreb, in 2019, where he is pursuing the Ph.D. degree. He is currently working as a Young Researcher with the Image Processing Group, Department of Electronic Systems and Information Processing. His research interests include visual quality control, deep learning object detection, and synthetic image generation.

Marko Subašić (Member, IEEE) received the Ph.D. degree from the Faculty of Electrical Engineering and Computing, University of Zagreb, in 2007. Since 1999, he has been working at the Department for Electronic Systems and Information Processing, Faculty of Electrical Engineering and Computing, University of Zagreb, where he is currently working as an Associate Professor. He teaches several courses at the graduate and undergraduate levels. His research interests include image processing and analysis and neural networks, with a particular interest in image segmentation, detection techniques, and deep learning. He is a member of the Croatian Center for Computer Vision, the Croatian Society for Biomedical Engineering and Medical Physics, and the Centre of Research Excellence for Data Science and Advanced Cooperative Systems.

Marko Budimir (Member, IEEE) received the M.Sc. degree in physics from the Faculty of Science, University of Zagreb, in 2000, and the Ph.D. degree from the Ecole Polytechnique Federale de Lausanne, Switzerland, in 2006. From 2006 to 2008, he worked at EPFL. Since 2008, he has been working at the Institute of Nuclear Technology (INETEC). He coordinated many key projects at INETEC and although he is a key person in a company of industry sector, he is still working close to the field of science.

Sven Lončarić (Senior Member, IEEE) received the Doctor of Philosophy (Ph.D.) degree in electrical engineering from the University of Cincinnati, USA, in 1994. With his students and collaborators, he coauthored more than 200 publications in scientific journals and conferences. He is the Founder of the Center for Computer Vision, University of Zagreb. He is the Head of the Image Processing Group. He is a Full Professor of Electrical Engineering and Computer Science with the Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia. He has served as the Co-Director of the National Center of Research Excellence in Data Science and Cooperative Systems. He was the Chair of the IEEE Croatia Section. He is a member of Croatian Academy of Technical Sciences. He received several awards for his scientific and professional work.

Biography

Duje Medak was born on December 5th, 1995. in Split, Croatia, where he attended primary and secondary school after which he continued his education at the University of Zagreb, Faculty of Electrical Engineering and Computing, where he received his M.Sc. in 2019. Since 2018., he has been working as a Young researcher at the Department of Electronic Systems and Information Processing, Faculty of Electrical Engineering and Computing, University of Zagreb where he participated in the scientific project KK.01.2.1.01.0151 (Smart UTX). His research interests include image processing, image analysis, machine learning, and deep learning. He is involved in educational activities on courses Digital Logic. He was also involved in the organization of several international conferences, workshops, and summer schools. He is an author or co-author of six journal papers and three conference papers.

Published articles

Journal publications

1. **D. Medak**, L. Posilović, M. Subašić, M. Budimir, S. Lončarić, "Automated Defect Detection from Ultrasonic Images Using Deep Learning", *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, Vol. 68, No. 10, 2021, pp. 3126-3134.
2. **D. Medak**, L. Posilović, M. Subašić, M. Budimir, S. Lončarić, "DefectDet: a deep learning architecture for detection of defects with extreme aspect ratios in ultrasonic images", *Neurocomputing*, vol. 473, Feb. 2022, pp. 107-115.
3. **D. Medak**, L. Posilović, M. Subašić, M. Budimir, S. Lončarić, "Deep learning-based defect detection from sequences of ultrasonic B-scans", *IEEE Sensors*, vol. 22, no. 3, Feb. 2022, pp. 2456-2463.

4.L. Posilovi ć, **D. Medak**, M. Subašić, M. Budimir, S. Lončarić, "Generative adversarial network with object detector discriminator for enhanced defect detection on ultrasonic B-scans", *Neurocomputing*, Vol. 459, Oct. 2021, pp. 361-369.

5.L. Posilovi ć, **D. Medak**, M. Subašić, M. Budimir, S. Lončarić, "Generating ultrasonic images indistinguishable from real images using Generative Adversarial Networks", *Ultrasonics*, Vol. 119, Feb 2022, pp. 361-369.

6.L. Posilovi ć, **D. Medak**, F. Milkovic, M. Subašić, M. Budimir, S. Lončarić, "Deep learning-based anomaly detection from ultrasonic images", *Ultrasonics*, vol. 124, August 2022.

Conference publications

1. **D. Medak**, L. Posilović, M. Subasić, T. Petković, M. Budimir and S. Lončarić, "Rapid Defect Detection by Merging Ultrasound B-scans from Different Scanning Angles", in *12th International Symposium on Image and Signal Processing and Analysis (ISPA)*, 2021, pp. 219-224
2. L. Posilovi ć, **D. Medak**, M. Subašić, T. Petković, M. Budimir and S. Lončarić, "Synthetic 3D Ultrasonic Scan Generation Using Optical Flow and Generative Adversarial Networks", in *12th International Symposium on Image and Signal Processing and Analysis (ISPA)*, 2021, pp. 213-218
3. L. Posilovi ć, **D. Medak**, M. Subašić, T. Petković, M. Budimir, S. Lončarić, "Flaw Detection from Ultrasonic Images using YOLO and SSD", in *11th International Symposium on Image and Signal Processing and Analysis (ISPA)*, 2019, pp. 163-168.

Životopis

Duje Medak rođen je 5. prosinca 1995. u Splitu, Hrvatskoj. Tamo je pohađao osnovnu i srednju školu, a obrazovanje je nastavio na Sveučilištu u Zagrebu, na Fakultetu elektrotehnike i računarstva, gdje je diplomirao 2019. godine. Od 2018. radi kao Mlađi istraživač na projektu SmartUTX (KK.01.2.1.01.0151) na Zavodu za elektroničko inženjerstvo i obradbu informacija na Fakultetu elektrotehnike i računarstva Sveučilišta u Zagrebu. Njegovi interesi uključuju obradu i analizu slike, strojno učenje i duboko učenje. Uz istraživački projekt radi i kao asistent u nastavi na predmetu Digitalna logika. Također, bio je uključen u organizaciju nekoliko međunarodnih konferencija, radionica i ljetnih škola. Autor je i koautor šest članka u časopisu i tri konferencijska članka.