Deep learning-based analysis of fuel spray images

Huzjan, Fran

University of Zagreb

FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

Fran Huzjan

DEEP LEARNING-BASED ANALYSIS OF FUEL SPRAY IMAGES

DOCTORAL THESIS

Zagreb, 2023

University of Zagreb

FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

Fran Huzjan

DEEP LEARNING-BASED ANALYSIS OF FUEL SPRAY IMAGES

DOCTORAL THESIS

Supervisor: Academic Professor Sven Lončarić, F.C.A

Zagreb, 2023

Sveučilište u Zagrebu
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Fran Huzjan

ANALIZA SLIKA SPREJA GORIVA DUBOKIM UČENJEM

DOKTORSKI RAD

Mentor: akademik prof. dr. sc. Sven Lončarić

Zagreb, 2023.

This doctoral thesis was completed at the University of Zagreb Faculty of Electrical Engineering and Computing, Department of Electronic Systems and Information Processing. This research was funded by the European Regional Development Fund, Operational Programme Competitiveness and Cohesion 2014-2020, KK.01.1.1.04.0070.

Supervisor: Academic Professor Sven Lončarić, F.C.A

This dissertation has: 77 pages

Dissertation number: _____

O mentoru

Sven Lončarić diplomirao je i magistrirao u polju elektrotehnike na Fakultetu elektrotehnike i računarstva, 1985. i 1989. godine. Doktorirao je u polju elektrotehnike na Sveučilištu u Cincinnatiju, SAD, 1994. godine. U zvanje redoviti profesor u trajnom zvanju u polju elektrotehnike i polju računarstva na FER-u izabran je 2011. godine. Bio je suradnik ili voditelj na brojnim istraživačkim i razvojnim projektima u području razvoja metoda za obradu slika i računalnog vida. Od 2001. do 2003. bio je Assistant Professor na Sveučilištu New Jersey Institute of Technology, SAD. Voditelj je istraživačkog laboratorija za obradu slike na FER-u. Osnivač je i voditelj Centra izvrsnosti za računalni vid na Sveučilištu u Zagrebu. Suvoditelj je nacionalnog Znanstvenog centra izvrsnosti za znanost o podatcima i kooperativne sustave i voditelj Centra za umjetnu inteligenciju FER-a. Sa svojim studentima i suradnicima publicirao je više od 250 znanstvenih i stručnih radova. Prof. Lončarić redoviti je član Hrvatske akademije znanosti i umjetnosti. Prema studiji Sveučilišta Stanford objavljenoj 2022. godine rangiran je u 2% najutjecajnijih svjetskih znanstvenika u kategoriji umjetna inteligencija i obrada slike. Za svoj znanstveni i stručni rad dobio je više nagrada uključujući Državnu nagradu za znanost.

About the Supervisor

Sven Lončarić received a Diploma of Engineering and Master of Science degrees in electrical engineering from the Faculty of Electrical Engineering and Computing in 1985 and 1989, respectively. He received a Ph.D. degree in electrical engineering from the University of Cincinnati, USA, in 1994. Since 2011, he has been a tenured full professor in electrical engineering and computer science at FER. He was a project leader on a number of research projects in the area of image processing and computer vision. From 2001-2003, he was an assistant professor at New Jersey Institute of Technology, USA. He founded the Image Processing Laboratory at FER and the Center for Computer Vision at the University of Zagreb. Prof. Lončarić has been a co-director of the National Center of Research Excellence in Data Science and Cooperative Systems and the director of the Center for Artificial Intelligence at FER. With his students and collaborators, he published more than 250 scientific papers. Prof. Lončarić is a Fellow of the Croatian Academy of Sciences and Arts. According to a Stanford University study published in 2022, he was ranked in the top 2% of the most cited world scientists in the category of artificial intelligence – image processing. He received several awards for his scientific work, including the National Science Award.

Preface

I wish to express my profound gratitude to my thesis supervisor, Prof. Sven Lončarić, for his invaluable guidance and unwavering support throughout the development of this thesis. His contributions to the scientific papers contained within this work are deeply appreciated. I also extend my heartfelt thanks to Prof. Marko Subašić for his selfless assistance and insightful direction in my research journey.

I am equally grateful to Prof. Milan Vujanović, the project leader of "RESIN", and his esteemed colleague from the Faculty of Mechanical Engineering and Naval Architecture, Filip Jurić, Ph.D. Their immense help throughout the project and beyond has been instrumental in its success.

I extend my special appreciation to my parents, Arijana and Boris. Their constant encouragement has been my rock, shaping me into the person I am today. Equally, I owe a debt of gratitude to my amazing sister Petra, my beloved family, my best office in the world D159, my dog Rocky, and my close friends. Their reassurance and cheer throughout my doctoral study and educational journey have been invaluable. The path toward this Ph.D. would have been far more challenging without their support.

Lastly, the most heartfelt "Thank You" is reserved for my best girlfriend, Barbara. Your enduring patience and motivation, particularly through the most challenging times, have been my anchor. Your unwavering belief in my abilities, encapsulated in your encouraging words - "you can do it", have consistently inspired me to strive for excellence. I am profoundly grateful for your love and support.

Abstract

Internal combustion engines with direct rail injection systems rely on spray strategies to improve engine efficiency, enhance the combustion process, and reduce the formation of pollutants. These spray strategies are determined by a variety of parameters, including the nozzle diameter, injection pressure, injector geometry, and cylinder type. Together, these factors impact the shape of the spray, which in turn influences the spray's macroscopic parameters. In numerical simulations, the spray macroscopic parameters are frequently used to describe the input parameters. These parameters include the spray cone angle, penetration length, and spray area. The spray cone angle refers to the width of the spray as it emerges from the nozzle, while the penetration length represents the distance the spray travels before it begins to disperse. The spray area is the total surface area covered by the spray. Overall, optimizing spray strategies and their associated macroscopic parameters is critical to achieving efficient, clean combustion in internal combustion engines with direct rail injection systems. By carefully controlling these parameters, engineers can improve engine performance, reduce emissions, and create a more sustainable future for transportation. To derive spray macroscopic parameters from spray images, computer vision methods were employed. While many existing methods in computer vision literature for spray image analysis are non-learning based, recent advancements in GPU technology have made it possible to develop more sophisticated algorithms based on artificial intelligence and deep learning techniques. Thus, learning-based approaches were utilized to accurate and efficiently obtain spray parameters. These techniques have been demonstrated to achieve superior results in numerous domains and offer a promising avenue for future research. In this thesis, two distinct methods were utilized to obtain macroscopic parameters from spray images. The first method employed image segmentation techniques to estimate parameters from the segmented image, while the second method directly estimated the parameters from the sequence of images itself. To facilitate the first method, a lightweight segmentation neural network, named Min U-Net, was developed. Min U-Net demonstrated exceptional results, performing comparable results as the other state-of-the-art segmentation models while having significantly fewer parameters and being much faster. For the second method, a regression deep neural network was utilized, consisting of a feature extractor and classifier. Notably, this method incorporated a sequence of images, rather than a single image, resulting in more accurate parameter estimates than the single-image state-of-the-art methods. These novel methods offer promising avenues for further research in spray analysis and may prove useful in optimizing spray strategies for improved engine efficiency and reduced pollutant emissions.

Keywords: Computer Vision, Artificial Intelligence, Neural Networks, Deep Learning, Image Segmentation, Image Analysis, Regression, Spray, Diesel, Macroscopic Parameters

Prošireni sažetak

Iako je razvoj motora s unutarnjim izgaranjem za osobna vozila značajno usporen zbog sve veće popularnosti elektrifikacije, on ostaje ključan za industriju teškog transporta. Glavni fokus u smanjenju štetnih emisija jest uvođenje ugljično neutralnijih goriva, poput biogoriva i e-goriva. Kako bi se proučila svojstva spreja i izgaranja ovih alternativnih goriva i njihovih mješavina s tradicionalnim gorivima, koriste se kombinirana eksperimentalna i numerička istraživanja. Ubrizgavanje goriva predstavlja ključni proces koji utječe na performanse izgaranja motora. Strategije ubrizgavanja imaju direktan utjecaj na učinkovitost motora, proces izgaranja i smanjenje onečišćenja u motorima s unutarnjim izgaranjem koji koriste sustave izravnog ubrizgavanja. Oblik spreja goriva određuje nekoliko parametara, uključujući promjer mlaznice, tlak ubrizgavanja, geometriju injektora i vrstu cilindra, koji utječu na makroskopske parametre spreja. Makroskopski parametri spreja, poput kuta spreja, penetracije spreja i površine spreja, uobičajeno se koriste za opisivanje ulaznih parametara numeričkih simulacija. Duljina penetracije spreja određuje se kao ukupna udaljenost koju sprej prelazi od mlaznice duž središnje osi. Penetracija spreja pomaže u otkrivanju sudara sa stijenkama cilindra, što utječe na učinkovitost izgaranja. Što je duljina penetracije veća, veći je udar na zid cilindra. Kut spreja ima mnogo definicija, ali za jednostavnost, možemo reći da je to kut između dviju ravnih linija koje počinju na mlaznici i dodiruju rubove konture spreja. Kut spreja utječe na procese potrošnje zraka i miješanja. Površina spreja određuje ukupni prostor koji zauzima raspršeno gorivo. Veća površina spreja rezultira većim kontaktom između zraka i goriva, što poboljšava brzinu isparavanja tijekom izgaranja. Jedan od najčešćih pristupa za optimizaciju sustava ubrizgavanja spreja je numerička simulacija s računalnom dinamikom fluida (CFD), koja nudi niže troškove u odnosu na eksperimentalna istraživanja. Međutim, unatoč širokoj primjeni CFD simulacija, one i dalje ovise o eksperimentalnim istraživanjima za provjeru valjanosti CFD modela, njihovih ulaznih parametara i početnih uvjeta. Zbog toga je nužno optički mjeriti svojstva spreja kako bi se dobila precizna CFD simulacija spreja. Dominantna metoda za određivanje makroskopskih parametara uključuje optičko mjerenje u kombinaciji s analizom slike. Razne tehnike se primjenjuju za optička mjerenja, poput dijagrama sjene, Schlieren fotografije, tehnika raspršenja, laserski inducirane fluorescencije, balističkog snimanja i rendgenskog snimanja. Nakon snimanja slika velikom brzinom, koriste se algoritmi za obradu slike kako bi se odredili globalni parametri spreja. Glavni izazov kod algoritama iz literature za procjenu parametara spreja jest što se svaki hiperparametar algoritma mora ručno podešavati za pojedinačne slike spreja kako bi se postigla optimalna točnost. Loše ručno podešavanje hiperparametara može dovesti do netočne procjene i potencijalno većeg onečišćenja s nižom učinkovitošću motora. Umjetna inteligencija i metode učenja značajno su unaprijedile područje računalnog vida. Primjena dubokih neuronskih mreža proširila se u mnogim područjima, čime je njihova popularnost učinila ih

izvrsnim rješenjem za problem procjene makroskopskih parametara spreja s fotografija. Ova doktorska disertacija istražuje područje koje kombinira umjetnu inteligenciju i metode procjene makroskopskih parametara s fotografija ubrizgavanja spreja goriva, s ciljem poboljšanja točnosti i učinkovitosti analize i optimizacije sustava ubrizgavanja.

Prvo poglavlje uvodi čitatelja u temu disertacije, pružajući opis problema i širi kontekst istraživanja. Detaljno se objašnjava motivacija za primjenu neuronskih mreža u procjeni makroskopskih parametara spreja. Također, iznose se znanstveni doprinosi doktorske disertacije te se daje kratak pregled načina na koji su postignuti. Fokus doprinosa leži u razvoju modela dubokog učenja za procjenu parametara spreja i rješavanje povezanih problema. Na kraju poglavlja, predstavlja se struktura disertacije i njezinih dijelova.

Drugo poglavlje detaljno obrađuje osnove računalnog vida, uključujući analizu slika i analizu slika spreja. Predstavljene su temeljne ideje neuronskih mreža, aktivacijske funkcije koje se koriste u njima, te revolucionarna arhitektura koja je preobrazila računalni vid - konvolucijske neuronske mreže. Opisani su najpoznatiji skupovi podataka, kao što su ImageNet, Kitti, Pascal i CityScapes, koji su bili ključni za napredak računalnog vida. Objašnjava se načelo učenja neuronskih mreža, kao i važnost računalne snage u tom procesu. Pojam semantičke segmentacije razrađuje se, zajedno s njezinim najčešćim primjenama i problemima s kojima se susreće, poput veličine objekata na slici, veličine jezgre konvolucijskog sloja te vrste enkodera i dekodera u neuronskim mrežama. U nastavku drugog poglavlja detaljno se opisuju konvolucijske neuronske mreže, počevši od jednostavnog višeslojnog perceptrona koji je poslužio kao temelj za razvoj potpuno povezanih i konvolucijskih neuronskih mreža. Uobičajeni slojevi u konvolucijskim mrežama, uz konvolucijske slojeve, uključuju slojeve sažimanja. Kombinacijom navedenih i prethodno spomenutih aktivacijskih funkcija, među kojima je najpopularnija ReLU aktivacijska funkcija, konvolucijske neuronske mreže stvaraju mapu značajki. Mapa značajki predstavlja 2D matricu koja sadrži informacije izvučene iz ulazne slike putem aktivacije određenih filtera koji se primjenjuju na specifične regije. Kasnije u drugom poglavlju, daje se pregled aktualnih metoda iz literature koje se koriste za analizu slika spreja. Najčešće metode koje se koriste za prikupljanje slika sprejeva uključuju sustav za snimanje sjena, Schlieren snimanje te snimanje temeljeno na laseru. Te metode omogućuju dobivanje slika spreja koje se zatim analiziraju pomoću tradicionalnih algoritama, kao što su detekcija rubova, segmentacija pragom, Otsu segmentacija i slično. Različiti algoritmi daju različite rezultate ovisno o svojim podesivim parametrima, poznatim kao hiperparametri. Ručno podešavanje hiperparametara pokazalo se kao glavni problem u generalizaciji, odnosno otpornosti algoritma na različite vrste slika spreja. Na kraju drugog poglavlja, opisuju se makroskopski parametri spreja i njihov utjecaj na proces izgaranja, učinkovitost motora i onečišćenje zraka. Kroz cjelokupnu disertaciju, istraživanje se usredotočuje na kombiniranje umjetne inteligencije i metoda estimacije makroskopskih parametara sa slika ubrizgavanja spreja goriva, s ciljem poboljšanja točnosti i

generalizacije tih metoda. Time se nastoji unaprijediti performanse motora, smanjiti zagađenje i pridonijeti razvoju održivijih tehnologija u sektoru prometa i prijevoza.

Treće poglavlje se fokusira na analizu i pripremu podataka koji se koriste za učenje i evaluaciju metoda ispitanim i ostvarenim u ovoj doktorskoj disertaciji. Slike su prikupljene od istraživačke grupe iz Poljske, sa instituta Techniki Cieplnej. Gorivo je ubrizgano pod tlakom od 32 MPa u posudu s konstantnim tlakom od 3.2 MPa. Slike su snimane visokofrekventnom kamerom Photron SA1.1. Za cijeli proces ubrizgavanja, određene su vrijednosti makroskopskih parametara spreja za specifičan skup slika. Raspon kuta spreja nalazi se između 17.25 i 26.03 stupnjeva, s prosječnom vrijednošću od 21,41 stupnja. Penetracija spreja varira od 114.0 do 570.0 piksela, s prosječnom vrijednošću od 479.10 piksela. Površina spreja prosječno iznosi 59.557.62 kvadratnih piksela, a nalazi se u rasponu od 2663.0 do 74912.0 kvadratnih piksela. Na slikama, sprej je prikazan svijetlo smeđe boje, dok je pozadina tamnije smeđe nijanse. Jedan od dodatnih detalja koji se može primijetiti na slikama je lampa smještena na desnoj strani slike. Za izračunavanje vrijednosti makroskopskih parametara spreja, koje se koriste kao referentne vrijednosti u daljnjoj evaluaciji i učenju metoda dubokog učenja, korištene su metode iz literature koje se primjenjuju na segmentirane slike. Precizne segmentacijske maske označili su četiri stručnjaka za sprejeve, te su one poslužile kao osnova za izračunavanje parametara. Algoritmi iz literature temelje se na detekciji rubova i interpolaciji linija, što omogućuje precizno izračunavanje makroskopskih parametara spreja za daljnju analizu i usporedbu s rezultatima dobivenim iz metoda dubokog učenja.

U četvrtom poglavlju, iznosi se opis prvog doprinosa ovog disertacijskog rada, koji se odnosi na procjenu makroskopskih parametara spreja temeljenih na segmentaciji. Dosadašnje segmentacijske metode u literaturi koje određuju parametre spreja istaknule su potrebu za razvojem segmentacijskog modela dubokog učenja s niskom složenošću. To bi omogućilo postizanje veće točnosti generalizacije uz smanjenje potrebe za ručnim podešavanjem hiperparametara. U radu je opisana augmentacija skupa podataka koja koristi geometrijske transformacije nad slikama, poput rotacija i skaliranja. Augmentacija efikasno povećava veličinu i raznolikost skupa podataka za red veličine koji je jednak broju epoha tijekom kojih se model trenira. Skup podataka je podijeljen na skupove za treniranje, testiranje i validaciju. Predstavljena je metrika koja se koristi za evaluaciju razvijenih modela - Dice koeficijent. Iako je Dice sličan omjeru presjeka i unije (IoU), služi kao F1 mjera jer uzima u obzir prave pozitivne, lažne pozitivne i lažne negativne vrijednosti. Zbog svoje robustnosti i efikasnosti, Dice koeficijent je odabran kao metrika. U radu je predložena metoda Min U-Net, koja se temelji na suvremenom U-Netu, koji se pokazao kao izvrsna arhitektura za razne probleme segmentacije. Min U-Net se sastoji od enkodera i dekodera, ali za razliku od U-Neta ima manju dubinu arhitekture i manje konvolucijskih jezgri u konvolucijskim slojevima. Sadrži dva sloja za enkodiranje i dva sloja za dekodiranje. Nakon segmentacije slike, prethodno spomenute metode literature koriste se za

izračunavanje makroskopskih parametara spreja iz segmentirane slike. Da bi se pronašla optimalna arhitektura Min U-Neta, provedena je studija ablacije. U studiji su ispitane veličine jezgri u konvolucijskim slojevima, kao i dubina mreže, odnosno broj konvolucijskih slojeva. Budući da arhitektura Min U-Neta slijedi pravilo U-Neta, broj konvolucija se udvostručuje u svakom sljedećem konvolucijskom sloju. Napravljen je kartezijev produkt svih kombinacija modela s dubinama 1, 2, 3, 4 i 5 te početnim brojem konvolucija 1, 2, 4, 8, 16 i 32. Razvijeni modeli uspoređeni su na temelju Dice koeficijenta i broja parametara. Primijećeno je da što je model manje složen, to je manje točan, ali postoji trenutak kada dodatni parametri i složenost modela prestanu doprinositi točnosti. Uzimajući u obzir te informacije, četiri modela su dodatno uspoređena pomoću srednje vrijednosti, medijana, najboljih 25%, najgorih 25% i trimeana. Na temelju tih rezultata, arhitektura s dubinom 3 i početnim brojem konvolucija 4 pokazala se kao najbolji kandidat. Min U-Net uspoređen je s tradicionalnim metodama segmentacije, kao što su segmentacija pragom, Otsu segmentacija i segmentacija maksimalne entropije. Uspoređivanje je provedeno na slikama bez preprocesiranja, kao i na preprocesiranim slikama. Preprocesirane slike korištene su kako bi se olakšalo tradicionalnim metodama, ali u oba slučaja Min U-Net pokazuje bolju točnost od navedenih metoda koje nisu temeljene na učenju. Min U-Net također je uspoređen s drugim suvremenim arhitekturama, kao što su PSPNet, FPN, Linknet i U-Net. Svaka od arhitektura testirana je s okosnicama koje su Densenet, Dpn, EfficientNet, MobileNet, ResNet i VGG. Min U-Net postiže usporedive rezultate s navedenim modelima, ali to čini s mnogo manje parametara. Ima oko 500 puta manje parametara od najmanje složenog modela i preko 5600 puta manje parametara od najkompleksnijeg modela. Također, zbog svoje niske složenosti, Min U-Net segmentira sliku otprilike dvostruko brže nego najbrži testirani model, što pokazuje njegovu efikasnost. Na kraju poglavlja, makroskopski parametri spreja izračunavaju se iz segmentirane slike Min U-Neta, a ti rezultati uspoređuju se s rezultatima procjene parametara s segmentirane slike tradicionalnih metoda. Min U-Net računa kut spreja s pogreškom od 1.08 stupnjeva, što je dvostruko manja pogreška od sljedećeg najtočnijeg algoritma, segmentacija pragom. Za duljinu penetracije spreja postiže relativnu pogrešku od 5.95%, što je više od 4 puta točnije od sljedeće najtočnije metode, a za površinu spreja postiže relativnu pogrešku od 4.05%, dok segmentacija pragom, koja se pokazala kao najtočnija među tradicionalnim metodama, postiže veću pogrešku od 14.36%.

Peto poglavlje pruža pregled drugog doprinosa ovog doktorskog rada, koji se temelji na procjeni makroskopskih parametara pomoću regresijskih dubokih modela, koristeći sekvencu slika umjesto pojedinačne slike, kako je to učinjeno u prethodnim pristupima. Razlog korištenja više slika je taj što se s obzirom na to da su slike prikupljene tijekom jednog ubrizgavanja goriva, nekoliko prethodnih slika može iskoristiti za točniju procjenu parametra posljednje slike spreja. Prvo su obrađene ulazne slike tako da se svaka slika postavi na isti način. Svaki sprej na slici zarotiran je tako da se širi s lijeva na desno te je smješten točno na sredini visine slike, čime je

postignuta uniformna augmentacija. Budući da se za ovaj eksperiment koristilo samo 196 slika spreja, augmentacija je bila ključna. Upotrebljavane su dvije vrste augmentacija: prije početka treninga i tijekom dohvaćanja slika za treniranje. Augmentacija slika spreja prije treniranja ostvarena je tako da se preprocesirane slike spreja šire i sužavaju, postižući različite vrijednosti kuteva spreja. Augmentacijom je efektivni broj slika spreja povećan s 196 na 3276 slika, povećavajući standardnu devijaciju i raspon kuteva. Predložene su dvije metode, StackNet i CNN-LSTM, koje se temelje na regresijskim dubokim neuronskim mrežama. Obje metode koriste ekstraktor značajki i potpuno povezan sloj na kraju modela, s razlikom što CNN-LSTM koristi dodatni LSTM sloj. Oba modela evaluirana su s tri vrste ekstraktora značajki: VGG, EfficientNet i MobileNet. Kao nadogradnja na StackNet, predložena je nova arhitektura pod nazivom Extended StackNet. Razlika između StackNeta i Extended StackNeta je što prošireni StackNet uz ekstraktor značajki ima i potpuno povezan sloj koji na ulaz prima prethodne kuteve iz sekvence slika. Taj ulazni vektor kuteva enkodira se i zatim spaja s mapom značajki, te se to zajedno šalje u potpuno povezan sloj, odnosno regresor modela. Motivacija za paralelni dodatni potpuno povezan sloj je da dodatne informacije uz same slike mogu pružiti bolja i točnija rješenja. Testirane su četiri vrste proširenog StackNeta: Običan prošireni StackNet, prošireni StackNet s 3D konvolucijama, MiniEStackNet i MiniEStackNet16 koji je manje složen od MiniEStackNeta. Za evaluaciju modela korištena je srednja apsolutna pogreška, dok se za treniranje modela koristila srednja kvadratna greška zbog njene derivabilnosti. Modeli su testirani s brojem slika u sekvenci od 2 do 5 te su uspoređeni s modelima iste arhitekture, ali koji primaju samo jednu sliku. Modeli koji primaju samo jednu sliku vraćaju grešku između 1.173 i 2.515 stupnjeva. StackNet temeljen na VGG-u s ulaznom sekvencijom duljine tri slike pokazao se kao najbolji među CNN-LSTM i StackNet modelima, postižući grešku od 0.505 stupnjeva. VGG se pokazao kao najtočniji ekstraktor značajki s greškom od 0.982. Sekvencija duljine tri slike pokazala se kao najbolja za StackNet i CNN-LSTM, s prosječnom greškom od 1.097 stupnjeva. Prošireni StackNet pokazao se kao uspješan eksperiment, postižući još bolje rezultate od najboljeg tradicionalnog StackNeta. MiniEStackNet s duljinom sekvence od četiri slike postiže grešku od 0.476 stupnjeva. Kada se prosjek modela promatra preko svih duljina sekvenci slika, VGG-bazirani prošireni StackNet postigao je najmanju grešku od 0.528 stupnjeva. Za razliku od tradicionalnog StackNeta i CNN-LSTM-a, najmanja prosječna greška kada se promatra duljina sekvence postignuta je za duljinu sekvence od četiri slike, u iznosu od 0.610 stupnjeva. Istraživanje je pokazalo da paralelni sloj i njegovo enkodiranje kuteva doprinose točnosti izračuna makroskopskih parametara spreja, čak i uz korištenje manje parametara, što je ilustrirao MiniEStackNet s 2.476 milijuna parametara.

Šesto i posljednje poglavlje donosi zaključak istraživanja. Na temelju rezultata ovog istraživanja može se zaključiti da su predložene arhitekture neuralnih mreža uspješne i valjane. Prvi znanstveni doprinos pokazuje da segmentacija spreja s pomoću Min U-Net mreže pruža jednako

visoku točnost kao i najnaprednije neuronske mreže, ali s znatno manjim brojem parametara i složenošću. Također, rezultati ukazuju na to da Min U-Net nadmašuje tradicionalne metode koje nisu utemeljene na učenju. Min U-Net pruža i najbolje rezultate u procjeni makroskopskih parametara spreja temeljenih na segmentiranoj slici. Drugi znanstveni doprinos ističe da regresijski modeli StackNet i CNN-LSTM pružaju bolje rezultate kada se koristi sekvencija slika kao ulaz, u usporedbi s tradicionalnim pristupom u literaturi koji koristi samo jednu sliku. Upotrebom dodatnih podataka i njihovim enkodiranjem pomoću proširenog StackNet modela postignut je još precizniji pristup procjeni kuta spreja s pomoću sekvence slika. Sve predložene metode su u potpunosti automatizirane, reproducibilne te pružaju jednako dobre ili bolje rezultate od trenutnog stanja u literaturi. Ovim istraživanjem pokazano je da su razvijeni pristupi značajno unaprijedili točnost i učinkovitost u procjeni makroskopskih parametara spreja te predstavljaju značajan napredak u odnosu na dosadašnje metode.

Ključne riječi: Računalni vid, Umjetna inteligencija, Neuronske mreže, Duboko učenje, Segmentacija slika, Analiza slika, Regresija, Sprej, Dizel, Makroskopski parametri

Contents

Chapter 1

Introduction

1.1 Overview

The electrification of everyday vehicles has led to a significant reduction in the development of internal combustion engines for passenger cars. However, heavy-duty transport, responsible for 28% of the total carbon dioxide (CO2) emissions from the road transport sector despite only comprising 7% of the global vehicle fleet, still relies heavily on these engines [1]. In light of this, there is an urgent need to reduce the harmful emissions produced by these engines. One possible solution is the implementation of new fuels that are more carbon-neutral, such as biofuels and e-fuels. Investigating the spray and combustion properties of such fuels, as well as their blends with conventional fuels, requires a combination of experimental and numerical investigations [2]. Moreover, spray systems and injection strategies play a crucial role in optimizing engine efficiency, the combustion process, and pollutant formation within internal combustion engines [3].

Computational fluid dynamics (CFD) simulations are commonly used to optimize spray injection systems due to their lower cost compared to experimental investigations [4]. However, CFD simulations are still reliant on experimental research to validate CFD models and determine the exact numerical input conditions and parameters governing spray modeling approaches [5]. Therefore, it is essential to accurately measure spray properties to provide an accurate description of the spray in CFD simulations [6]. The spray cone angle and spray penetration are critical parameters that indicate the quality of vaporization, air-fuel mixing, ignition timing, combustion process, and pollutant formation inside the combustion chamber [7].

Overall, reducing the harmful emissions produced by internal combustion engines is crucial, especially for heavy-duty transport. Investigating new fuels and their blends, along with optimizing spray systems and injection strategies, can lead to significant improvements in engine efficiency and the reduction of pollutant formation.

The most effective way to obtain spray properties is by capturing the injection process with

a high-speed camera, followed by using image processing algorithms on the captured images to extract the spray macroscopic parameters. Currently, literature methods for obtaining these parameters primarily rely on traditional non-learning methods, such as thresholding techniques, including simple thresholding, Otsu, Max Entropy segmentation, and others [8, 9]. However, one of the main issues with this approach is that the segmentation parameters must be calculated or measured manually, which can be time-consuming and prone to errors. Additionally, this approach's thresholding method cannot guarantee that it will work equally well for all types of spray images.

The rise of convolutional neural networks (CNNs) and deep learning-based methods presents a viable solution to this problem, offering improved generalization, speed, and performance over traditional methods. However, these deep learning-based methods require large amounts of data for the learning process. Fortunately, several imaging techniques, such as Schlieren, elastic scattering, and Laser-Induced Fluorescence/Mie scattering, can obtain spray images, providing ample opportunities for the application of deep learning-based methods [10, 11]. Therefore, in this research, we aim to investigate the application of deep learning-based methods in the determination of spray macroscopic parameters, comparing their performance with traditional methods and exploring their potential for real-time implementation in combustion systems.

1.2 Scientific contributions

The present thesis aims to contribute to the field of spray image analysis by proposing deep learning-based methods to determine the spray macroscopic parameters. These parameters are crucial for understanding the combustion process in engines, as they impact the engine's performance, efficiency, and harmful emissions. Given the importance of these parameters for the environment and the performance of internal combustion engines, there is a pressing need for accurate and efficient methods to measure them. The traditional methods used in literature research for this purpose have limitations in terms of accuracy and automation. To overcome these challenges, this thesis explores the potential of deep learning methods to achieve more accurate and fully automated measurements of spray macroscopic parameters. Specifically, two deep learning-based methods are proposed in this thesis. The first method is a lightweight neural network based on the state-of-the-art U-Net architecture for segmentation. The second method is a regression neural network that utilizes a sequence of images as input to measure macroscopic spray parameters. The experimental results demonstrate the effectiveness of the proposed methods in achieving higher accuracy and greater automation compared to traditional methods. Thus, this thesis provides valuable contributions to the field of spray image analysis and highlights the potential of deep learning-based methods for advancing the understanding of the combustion process in engines. To conclude the scientific contributions of this thesis can be

summarized as:

1.Method for measurement of macroscopic spray parameters from a single image using a lightweight deep learning segmentation model

2.Method for measurement of macroscopic spray parameters from a sequence of images using a regression deep learning model

1.3 Organization of the thesis

The thesis is organized into six chapters, each addressing a specific aspect of the research. Chapter 1 introduces diesel spray fuels, their role in internal combustion engines, and the challenges associated with estimating spray macroscopic parameters. This chapter also outlines the scientific contributions of the thesis and provides an overview of the overall structure.

Chapter 2 serves as an introduction to computer vision, image analysis, neural networks, and semantic segmentation. It also offers a comprehensive review of the literature on spray image analysis and elaborates on the concept and definition of spray macroscopic parameters.

In Chapter 3, the process of data acquisition for the thesis is detailed, including the properties of the data and the preparation steps undertaken for the experiments.

Chapter 4 presents the development of a lightweight neural network for determining spray macroscopic parameters using segmentation. The first section explains the motivation behind this approach, while the second section discusses data augmentation. The third section focuses on the evaluation metrics for the results, and the fourth section describes the lightweight U-Net-based neural network. The fifth section elucidates the selection process of the neural network, and the final section analyzes the experimental results.

Chapter 5 introduces deep regression neural networks that utilize sequences of images as input. The first section provides the motivation for this approach, and the second section details data preprocessing and augmentation. The third section proposes the methods, and the fourth section presents the experimental results, comparing them with existing literature methods.

Lastly, Chapter 6 offers a conclusion to the thesis by providing a comprehensive overview of the developed methodology and summarizing the key findings.

Chapter 2

Computer vision, spray image analysis

2.1 Computer vision, image analysis, neural networks

Deep learning is a prominent field of machine learning that has garnered significant attention in recent years. It is particularly well-suited for problems pertaining to artificial intelligence, owing to its capacity to represent data with complex, learned nonlinear transformations. Among the many architectures of deep learning models, the most commonly used are deep neural networks, recurrent neural networks, convolutional neural networks, and their various combinations. Deep learning techniques have found applications in a diverse array of fields, including computer vision, speech recognition, natural language processing, bioinformatics, and others, as described in [12, 13, 14].

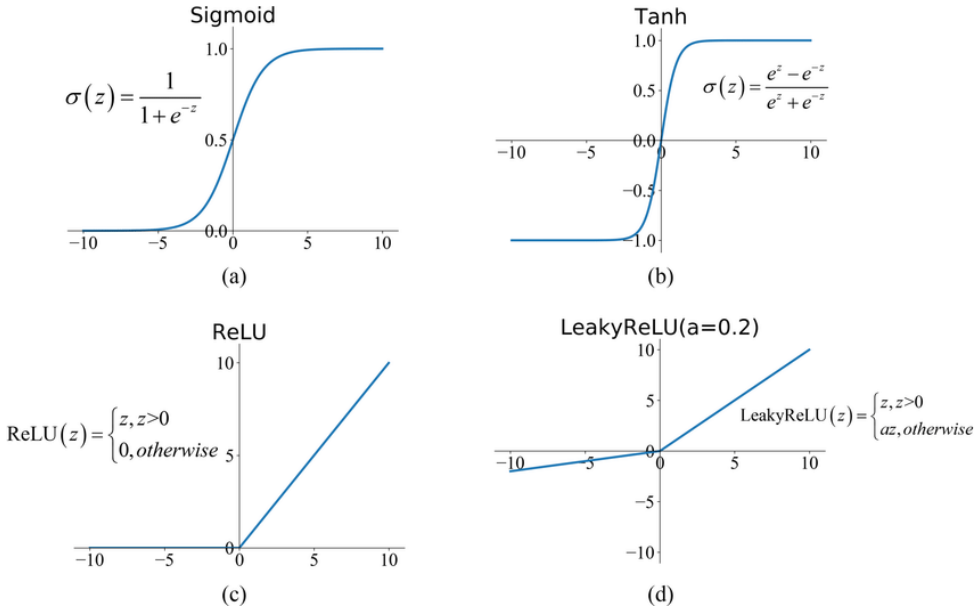

Figure 2.1: Most commonly used activation functions

Notably, deep learning has been at the forefront of many recent breakthroughs in computer vision. Convolutional neural networks (CNNs) are the primary deep neural network architecture

used in computer vision, as expounded in [15, 16]. Although CNNs have been used in computer vision for some time, the discovery of a novel activation function recently unlocked their full potential, as noted in [17]. Some of the most common activation functions are presented in Fig. 2.1. ImageNet, the first large-scale dataset in computer vision, is a widely adopted benchmark for evaluating the quality of CNNs, as outlined in [18]. Additional, smaller datasets, such as Pascal, Kitti, CityScapes, and others, have also been proposed, as discussed in [19, 20, 21]. GPU hardware has also played a significant role in the development of neural networks, as demonstrated in [22].

Deep learning models are typically composed of many layers, which means that they contain a vast number of parameters, typically between 10 and 50 million. Remarkably, state-of-the-art results have been achieved using models containing 557 million parameters on ImageNet, as described in [23]. Training deep learning models requires a significant amount of computational power, particularly with larger datasets. However, the amount of data present in a dataset is even more critical than computational power for training models. In most cases, supervised learning is used to train deep learning models. In supervised learning, data is annotated with labels by human experts in the relevant field of study. However, the process of annotating data can be expensive and time-consuming, leading to methods that rely on incomplete or non-existent annotations. Three forms of incomplete supervision are commonly used: weak supervision, semi-supervision, and noisy supervision, as discussed in [24, 25, 26].

2.1.1 Semantic Segmentation

Semantic segmentation is a crucial computer vision task that entails the generation of a new image in which every pixel is assigned to a particular class, such as road, vehicle, building, and so forth. The objective of semantic segmentation is to facilitate a computer's comprehension of an image at the pixel level, thus enabling it to recognize and differentiate between various elements within the image. The process of semantic segmentation has several applications, including medical image analysis, handwriting recognition, and self-driving cars, among others. For instance, semantic segmentation is a vital component of self-driving car technology, as it helps to detect obstacles and traffic, enabling the vehicle to take appropriate action based on the data.

The most commonly used neural network architecture for semantic segmentation involves multiple convolution layers, which are utilized to downsample the image. Downsampling acts as a noise suppressant, making the image invariant to translation movement, and capturing vital structural features of the image. In essence, the architecture extracts the features from the image, and this aspect is commonly referred to as the encoder. The encoder must be followed by the decoder, which upsamples the representation and reconstructs the original image from the extracted features. Additionally, the decoder computes class probabilities for each pixel using

the convolutional activation function, which is typically softmax [27].

Despite the utility of semantic segmentation, there are a few significant problems that arise in the process. One such issue is the challenge of recovering details for segmenting objects that are extremely small or large. When the model is segmenting large objects, it may need to look several hundred pixels away from the object to make the correct prediction. Conversely, when small objects with sizes as small as a few pixels are in question, downsampling causes the loss of almost all information about the objects. This makes it challenging for the model to rely solely on textures, and it needs a large receptive field to make the correct prediction.

To address the problem of segmenting extremely small objects, researchers have proposed the use of more sophisticated architectures that include pyramid pooling modules and a combination of convolutional layers with different kernel sizes. These techniques allow the network to capture multi-scale contextual information, enabling it to segment smaller objects accurately. Conversely, to segment large objects, researchers have proposed employing dilated convolutional layers or employing a spatial pyramid pooling module, which allows the network to capture features at multiple scales. These techniques have been found to be effective in segmenting objects of varying sizes [28].

2.1.2 Convolutional Neural Network

As already mentioned, the main network architecture for semantic segmentation is CNNs. CNNs are a class of deep neural networks based on Multi-Layer Perceptron (MLP) [29, 30]. They are also called fully connected feed-forward networks. The most basic element of MLP is a perceptron. A perceptron is a mathematical algorithm that maps out the input value to some output value based on the function which is built into it. An example scheme is shown in Fig. 2.2. In Fig. 2.2 x_i is the input, w_{ij} are the model's weights, b_j are the model's biases and φ is the activation function. The output y_j of the percepton is calculated as $y_j = \varphi(\sum_{i=1}^{n} w_{ij}x_i + b_j)$ Throughout the training process model's weights and biases are being learned.

When building the neural network, perceptrons are stacked into layers that are connected. In a fully connected network, every perceptron in one layer is connected with all perceptrons in the next layer. CNNs are similar to MLPs with the exception that there are convolution operations. The operations are built-in convolutional layers. Another exception is that in CNNs some perceptrons are only connected to a single perceptron in the next layer.

Convolutional layers can produce multiple independent outputs by using filters. Outputs are results of convolution of the entire image, also called feature maps. Convolutional layers are usually followed by pooling layers. The input to pooling layers is the small region of the feature map and the output is just one number determined by a function. Functions can be max function, also called max-pooling, or average value also called average pooling. Pooling is the downsampling of the input. Pooling aims to lessen the importance of the exact location

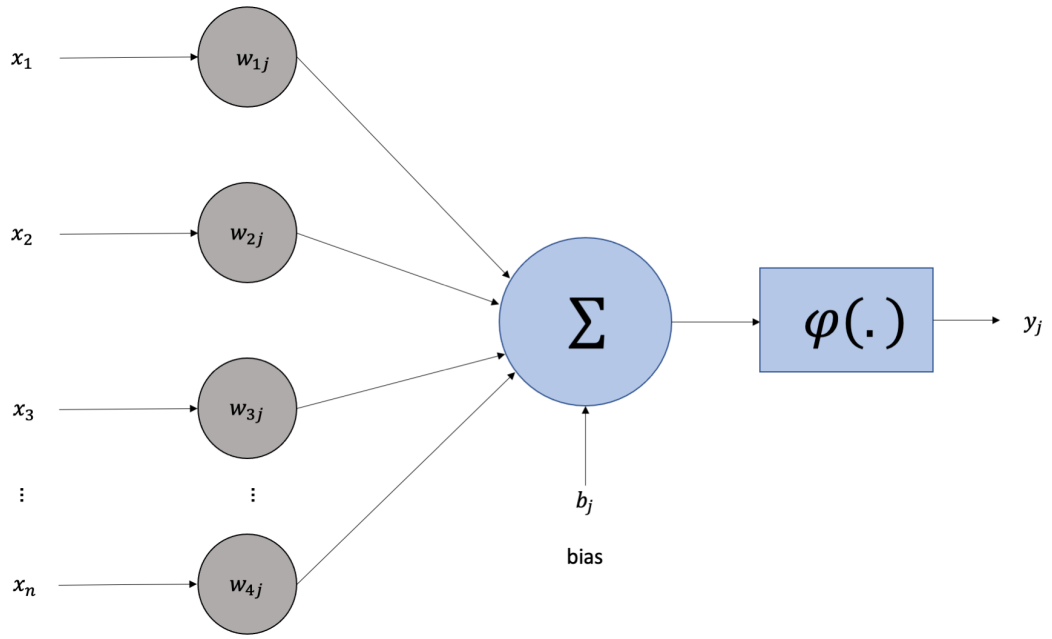

Figure 2.2: Perceptron scheme

of features and to reduce the number of parameters used to learn the network.

The design of convolutional neural networks for semantic segmentation is crucial for their performance. The two most common architectures are encoder-decoder and U-Net [31]. In the encoder-decoder architecture, the encoder part of the network extracts features from the input image and down-samples it. The decoder part of the network upsamples the feature map and constructs the output segmentation. U-Net architecture is similar to the encoder-decoder, but it has skip connections between encoder and decoder layers. Skip connections concatenate feature maps from the encoder to the decoder layer, allowing the decoder to take into account high-level features and detailed local information at the same time.

There are also variations of convolutional neural networks, such as dilated convolutions and pyramid pooling networks. Dilated convolutions allow the network to increase the receptive field of each neuron without downsampling. Pyramid pooling networks create multiple resolutions of the input image and extracts features from each resolution. Then the features are concatenated and used for segmentation. The main goal of these architectures is to improve the segmentation accuracy of the model.

The choice of activation function is also important in the design of the convolutional neural network. The most commonly used activation function is ReLU (Rectified Linear Unit), which is defined as $\varphi(\mathrm{x}) = \max(0, \mathrm{x})$. ReLU is computationally efficient and provides non-linear properties that allow the network to model complex non-linear relationships between input and output. Other activation functions that have been used in convolutional neural networks are sigmoid, tanh, and leaky ReLU.

CNNs for binary segmentation tasks, like spray segmentation, end with the final layer which gives the probability of each pixel for a given class, the background, or the spray in this case. A threshold is applied, most commonly it is 0.5 [32, 33].

2.2 Spray image analysis

Spray image analysis is a crucial aspect of investigating and characterizing the behavior of fuel sprays in diesel engines, as it provides valuable insights into the complex phenomena that govern the combustion process. Diesel engines operate on the principle of compression ignition, where fuel is injected as a high-pressure spray into the combustion chamber containing hot, compressed air. This process relies on several critical factors, including fuel atomization, droplet size distribution, penetration depth, spray angle, and evaporation rate.

Fuel atomization refers to the breakup of liquid fuel into small droplets, which is essential for creating a large surface area for the fuel to mix with air and evaporate. Achieving proper atomization ensures that the fuel burns more efficiently, leading to improved engine performance and reduced emissions. The droplet size distribution affects the rate of evaporation and combustion, as smaller droplets tend to evaporate more rapidly and burn more completely.

Given the importance of these factors in determining combustion efficiency, engine performance, and pollutant emissions, accurate analysis of diesel spray images is critical for optimizing the combustion process and designing more efficient and cleaner engines. By characterizing and understanding the complex interplay of fuel spray parameters, researchers and engineers can develop advanced injection strategies, optimize nozzle designs, and tailor fuel properties to achieve higher efficiency and lower emissions in diesel engines.

To study diesel sprays and understand their intricate characteristics, researchers have developed a variety of imaging techniques that capture and visualize the complex structures and dynamics of fuel sprays. These techniques enable the assessment of key parameters affecting the combustion process, such as droplet size distribution, spray penetration, and evaporation rate. Some of the most widely used methods include:

- Shadowgraphy: Shadowgraphy is an optical technique that leverages the differences in refractive index within the fluid to visualize spray structures [34]. It relies on the principle that light rays bend when they pass through regions of varying refractive indices, producing areas of light and shadow in the resulting image. This technique generates high-contrast images, which can be used to study spray penetration, overall shape, and droplet distribution. Shadowgraphy is particularly useful for studying transient processes, such as the initial stages of spray development, the interaction between multiple sprays, and the impact of ambient conditions on spray behavior.
- Schlieren Imaging: Schlieren imaging is another optical method that captures the changes

in refractive index caused by fluid density gradients [35]. This technique uses a specific optical setup, typically involving a light source, a collimating lens, a knife-edge or a focusing lens, and a camera. The setup is designed to enhance the visibility of density gradients, providing a clear view of the spray boundaries and enabling the analysis of spray angle, breakup, and vaporization. Schlieren imaging can reveal subtle features in the spray structure, such as shock waves and the formation of vortices, making it well-suited for studying high-pressure sprays and fuel-air mixing processes.

•Laser-based techniques: Laser-based methods, such as Laser-Induced Fluorescence (LIF) and Particle Image Velocimetry (PIV), use laser light to illuminate the spray, providing detailed information on various aspects of the spray.

- Laser-Induced Fluorescence (LIF) [36]: LIF is a non-intrusive technique that involves the excitation of fuel molecules or added tracers with a laser, causing them to fluoresce. The emitted fluorescence is then captured by a camera, providing information on the spatial distribution of fuel concentration, vaporization, and mixing with air. LIF can be used to study the temporal evolution of the spray and the effects of various parameters on the evaporation process.
- Particle Image Velocimetry (PIV) [37]: PIV is a technique that measures the velocity field of droplets or particles within the spray by capturing their displacement in successive images. The images are typically illuminated using a laser light sheet, and the droplet displacements are correlated between image pairs to calculate the droplet velocities. PIV enables the study of spray dynamics, turbulence, and the interaction between the spray and the surrounding air, providing valuable insights into the fuel injection and atomization processes.

These advanced imaging techniques, when combined with state-of-the-art image processing and analysis methods, can provide comprehensive information on the complex phenomena governing diesel spray behavior, contributing to the development of more efficient and cleaner combustion processes in diesel engines.

Various traditional image processing techniques have been employed to extract relevant information from spray images. These methods have served as the foundation for spray analysis and have been instrumental in developing our understanding of fuel spray behavior. These methods include:

•Edge Detection: Edge detection algorithms, such as Sobel, Canny, and Laplacian of Gaussian, are used to identify the boundaries of the spray and extract its overall shape. These techniques rely on the identification of abrupt changes in pixel intensity, which correspond to the edges of the spray. By isolating the spray boundaries, researchers can further analyze and calculate critical parameters like penetration depth, spray angle, and droplet distribution. These techniques, however, can be sensitive to noise and may require addi-

tional processing steps, such as filtering and smoothing, to achieve reliable results.

- •Thresholding: Thresholding techniques are essential for segmenting the spray region from the background, allowing for a more detailed analysis of spray characteristics, such as droplet distribution and vaporization. There are several thresholding methods, including:
  - Global Thresholding: This technique applies a single threshold value to the entire image, separating the spray from the background. The choice of the threshold value can be determined through various approaches, such as Otsu's method, which minimizes the intra-class variance of the segmented regions, or the Maximum Entropy method, which seeks to maximize the entropy between the foreground and background regions.
  - Adaptive Thresholding: Adaptive thresholding techniques, such as local or regional thresholding, compute a threshold value for each pixel based on the local neighborhood's characteristics. This approach is particularly useful when the image's lighting conditions vary or when the spray exhibits significant variations in intensity across its structure.
- •Morphological Operations: Morphological operations are essential for refining and enhancing spray boundaries, removing noise, and filling gaps in the spray structure. These operations are based on set theory and involve the manipulation of the spray's binary representation. Some common morphological operations include:
  - Erosion: Erosion shrinks the spray region by removing pixels from its boundaries. This process can be useful for eliminating noise, separating overlapping regions, and smoothing the spray's edges.
  - Dilation: Dilation expands the spray region by adding pixels to its boundaries. This operation can help fill gaps in the spray structure, connect disjointed regions, and reinforce weak edges.
  - Opening: Opening is a combination of erosion followed by dilation, which can remove small noise elements while preserving the spray's overall shape and size.
  - Closing: Closing is the reverse of opening, involving dilation followed by erosion. This process can help fill small gaps in the spray structure and smooth the boundaries without significant changes to the spray's size.

Different algorithms yield different values for spray parameters due to their inherent definitions and sensitivities to noise and image irregularities. In a review paper [38], the authors presented an overview of deep learning methods for multiphase interface detection, discussing the design choices of neural networks and their influence on dimension reduction techniques. They also concluded that deep learning methods offer considerable potential for advanced applications in stochastic multiphase problems, such as turbulence modeling in multiphase flows.

A spray processing method using a high-speed imaging system capable of obtaining schlieren and elastic scattering images through simple subtraction, filtering, and dilation was developed in [10]. The findings indicated that employing a small circular disc as a structuring element resulted in a higher discrepancy at peak liquid penetration. In [39], the hybrid image algorithm, which incorporated image subtraction operations between adjacent frames, was tested and demonstrated good agreement with one-dimensional validation models. For water sprays, image processing utilizing Laser-Induced Fluorescence/Mie scattering for droplet sizing was implemented to perform surface subtraction due to the strong reflection of laser scattering [11]. In the diluted spray region, approximately 300 diameters from the nozzle hole, Gaussian fitting with an intensity threshold was executed to calculate droplet-scale characterization [40]. The authors demonstrated that the combination of advanced imaging tools provides detailed insights into various spray multiphase scales. Digital image processing was subsequently conducted to determine tracer movements, which were used to obtain penetration profiles. Novel image processing software for optimizing spray parameters based on the Danielsson function and image mask for generating multiphase interfaces was developed in [41].

For the Engine Combustion Network Spray G validation case, an approach employing background subtraction and an optical thickness threshold was used, with the threshold set at 60% of pixel color extinction between the spray and background region [42]. A new image processing algorithm for determining diesel sprays was proposed, which divided each spray into three distinct regions and segmented each one with a separately calculated binarization threshold based on their luminosity profile [43].

The authors of [44] developed a method that eschewed image thresholding using the conventional Otsu method, instead estimating spray penetration and cone angle by comparing the original image with a generated synthetic shapes database. To validate the segmentation method, fluid transport in porous media was analyzed, revealing that image resolution was the most dominant factor in method accuracy [45]. The results indicated that image resolution affects the appearance of different phases and their interactions. An additional method for colored tracers in fluid was developed in [46].

While these traditional techniques can provide valuable insights, they often require manual tuning of parameters and can be sensitive to noise, variations in image quality, and changes in imaging conditions. Moreover, these methods may not capture complex and subtle features in the spray structure, limiting the accuracy and reliability of the analysis. The advent of deep learning-based approaches has offered new avenues for enhancing the precision and robustness of spray image analysis, enabling researchers to uncover previously hidden or challenging-to-detect patterns and features within the fuel spray.

2.3 Spray macroscopic parameters

In diesel engines, spray macroscopic parameters play a crucial role in determining the efficiency and performance of the combustion processes. A thorough understanding and optimization of these parameters are vital for designing more efficient and cleaner engines. This chapter delves into the importance of key spray macroscopic parameters, such as spray cone angle, penetration depth, and spray area, and their impact on fuel distribution, combustion quality, and pollutant emissions [47]. The spray macroscopic parameters can be seen on Fig. 2.3.

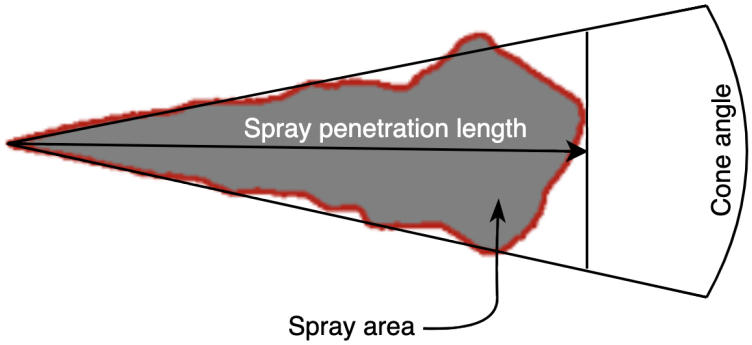

Figure 2.3: Macroscopic spray parameters definition

The spray cone angle is perhaps one of the most critical parameters, as it directly affects the distribution of fuel within the combustion chamber. The cone angle is defined as the angle between the outer edges of the spray cone. An optimal spray angle ensures proper fuel distribution within the combustion chamber, promoting efficient mixing with air and enabling effective combustion [48, 49]. Conversely, an incorrect spray angle can result in poor mixing and increased emissions of unburnt hydrocarbons and particulate matter. A narrower cone angle, for example, produces a more focused spray, which can lead to improved combustion efficiency and reduced pollutant formation. On the other hand, a wider cone angle may result in poorer combustion and increased emissions.

Penetration depth is another crucial parameter, measuring the distance the fuel spray reaches into the combustion chamber [48]. Adequate spray penetration is essential for ensuring thorough mixing of fuel and air, which, in turn, promotes efficient combustion. However, excessive penetration can lead to fuel impingement on the cylinder walls or piston, resulting in incomplete combustion and increased emissions. Longer penetration lengths can improve fuel distribution and enhance combustion efficiency, while shorter penetration lengths may result in uneven fuel distribution and reduced engine performance.

The spray area is an important parameter that provides a measure of the overall size of the spray. A larger spray area can improve fuel distribution and enhance combustion efficiency,

while a smaller spray area may lead to uneven fuel distribution and reduced performance [49]. An effective analysis of spray area can offer insights into the interaction between fuel droplets and air, as well as the overall homogeneity of the mixture within the combustion chamber. This information is crucial for optimizing the combustion process to minimize emissions and maximize engine performance.

In summary, spray macroscopic parameters, including spray cone angle, penetration depth, and spray area, play a pivotal role in the combustion process of diesel engines. A comprehensive understanding of these parameters and their interplay with fuel distribution, air-fuel mixing, and combustion quality is crucial for designing efficient and clean engines. This chapter provides an in-depth exploration of these parameters and their significance in optimizing diesel engine performance and reducing pollutant emissions.

Chapter 3

Data: Acquisition, properties, preparation of spray images

3.1 Acquisition

The spray data necessary for assessing the methods was procured by a research group from Instytut Techniki Cieplnej in Poland, who injected fuel into a constant volume chamber. The purpose of collecting this data was to create a model representing diesel fuel spray during the early phase of fuel spray within marine diesel engines. The chamber was filled with nitrogen at a pressure of 3.2 MPa, and fuel was injected at a pressure of 32 MPa through a nozzle hole with a diameter of 285 µm, located at the top of the chamber. A conventional pressure-opened diesel injector from a Sulzer Al 25/30 type marine engine, equipped with a Unit Pump injection system, was employed for this process. This experimental setup enabled the observation of a single fuel jet at a distance of approximately 100 mm from the spray axis.

The nozzle's design featured a cylindrical shape, with an orifice K factor of 1. The fuel pressure in the common rail system was maintained at 50 MPa, while the temperature of both the injected fuel and the chamber remained at approximately 22 degrees Celsius. Images were captured in RGB format at a resolution of 512x256 pixels using a Photron SA1.1 high-speed camera. In these images, one pixel is equivalent to 0.13 mm. The collected images had a frequency of 40 kHz, and the pressure within the injector was measured using a Kistler type 4067E piezoresistive pressure sensor.

3.2 Properties

Table 5.1 displays the metrics for spray macroscopic parameters obtained from the dataset. The cone angle degrees range between 17.25 and 26.03, with a standard deviation of 1.74, a mean of 21.41, and a median of 21.53. The spray penetration pixel length falls within an interval of

Table 3.1: Spray macroscopic parameters label metrics

	Mean	Median	Std. deviation	Max	Min
Cone angle (degrees)	21.41	21.53	1.74	26.03	17.25
Penetration length (pixels)	479.10	523.0	99.33	570.0	114.0
Spray area (pixels^2)	59557.62	67167.0	17961.95	74912.0	2663.0

114 to 570, featuring a standard deviation of 99.33, a mean of 479.10, and a median of 523. The spray pixel area values lie between 2663 and 74912, accompanied by a standard deviation of 17961.95, a mean of 59557.62, and a median of 67167.

In Figure 3.1, a representative image of the fuel spray within the constant volume chamber is displayed. The visualized spray is characterized by its light brown color, which extends from the top left corner towards the lower central region of the image. This spray pattern effectively demonstrates the dispersion and atomization of the diesel fuel within the chamber.

A notable feature in the image is the bright light source situated on the right-hand side. This illumination is intentionally employed to enhance the visibility of the fuel spray during the image acquisition process. By utilizing this light source, researchers can more accurately analyze the characteristics and behavior of the fuel spray under the given experimental conditions.

Figure 3.1: Example of the dataset spray image

3.3 Preparation

The original images were in Tag Image File Format (TIFF), a format commonly used for storing high-quality images. However, due to the substantial memory space requirements of TIFF images, they were converted to 8-bit Portable Network Graphics (PNG) format. PNG offers greater compatibility with deep learning libraries, smaller file sizes, and faster loading times. For utilization in convolutional neural networks, each image was resized individually.

Segmentation labels were acquired from four spray experts, who manually labeled the spray segmentation masks using the online tool Segments.ai. Figure 3.2 displays an example image and its corresponding mask. These segmentation masks were essential not only for the segmentation problem but also for obtaining ground truth labels. Since the dataset did not include ground truth values, these values were calculated using established literature methods for cone angle, spray penetration, and spray area. To obtain ground truth spray macroscopic parameters, the segmentation mask needs to be oriented correctly. The image's orientation angle was calculated using Principal Component Analysis (PCA), after which the image was rotated to the correct orientation.

Spray penetration was calculated as the difference between the x-coordinates of the first and last pixels with non-zero values. As the image was segmented, only two values were present: 0 for the background and 1 for the spray.

The literature method for cone angle determination is based on edge detection and line fitting [50, 51]. Acquiring edges on the segmented image is straightforward. The upper and lower edges up to half of the penetration were obtained, as current research in the literature suggests [52, 53, 54, 55]. Two lines, upper and lower, were fitted to the spray edges, minimizing the squared error between the fitted points and detected spray edges as shown in Eq. (3.1).

$$E = \sum_{j=1}^{k} p(x_j) - y_j{}^2 \tag{3.1}$$

where k is the number of elements used for spray angle calculation, x_j is the j^{th} point used for fitting, $p(x_j)$ is the j^{th} fitted point or predicted spray edge, and y_j is the j^{th} detected spray edge point.

The angle α between the upper and lower edge, $\vec{u}$ and $\vec{v}$ respectively, of the spray is calculated using Eq. (3.2).

$$\alpha = \arccos\left(\frac{\vec{u} \times \vec{v}}{\vec{u}\vec{v}}\right) \tag{3.2}$$

where $\vec{u}$ and $\vec{v}$ must be non-zero vectors; otherwise, no spray is detected in the image.

In 2-D, the equation for a line is defined in slope-intercept form as $y = mx + b$, where x, y are the Cartesian coordinates, and m and b are the line parameters, which can be any two real

Figure 3.2: Spray image (upper image) and it's corresponding segmentation mask (lower image)

numbers. The line fitting algorithm returns the slope m and the y-intercept b. To calculate the vector of the upper and lower edge, these parameters are used. In this method, the vector is calculated as follows:

$$\vec{a}x = S_{0.5x} - T_x, \tag{3.3}$$

$$\vec{a}_y = (m \times S_{0.5x} + b) - (m \times T_x + b), \tag{3.4}$$

$$\vec{a} = (\vec{a}_x, \vec{a}_y), \tag{3.5}$$

where $S_{0.5x}$ is the x coordinate at 0.5 times the length of spray penetration, T_x is the x coordinate where the spray penetration begins, and $\vec{a}_x$, $\vec{a}_y$ are the x and y components of the vector $\vec{a}$.

Using this approach, the macroscopic spray parameters, such as the cone angle, spray penetration, and spray area, are derived from the segmented images. These ground truth values, in conjunction with the segmentation masks, are crucial for training and evaluating the performance of deep learning models aimed at analyzing diesel fuel spray characteristics.

Chapter 4

Segmentation-based single image spray macroscopic parameters measurement

4.1 Motivation

The primary motivation for developing the proposed method stems from the increasing need for efficient and accurate solutions in the field of spray macroscopic parameter determination. The current state-of-the-art in this research domain predominantly relies on segmentation techniques, which have been found to yield varying degrees of success.

The objective of this study is to critically evaluate the performance of existing traditional methods in comparison with contemporary learning-based approaches and subsequently propose a novel, lightweight method for segmentation and parameter determination. Through a comprehensive review of the literature and a rigorous analysis of the underlying techniques, it has become evident that the current traditional methods possess considerable scope for improvement. Furthermore, it has been observed that state-of-the-art learning-based methods, while offering remarkable performance, tend to be overly complex and computationally intensive for the problem at hand.

In light of these findings, the development of a lightweight deep neural network assumes paramount importance. The proposed method is designed to be simple and computationally efficient while delivering results that are on par with the most advanced techniques available in the literature. By striking a balance between performance and resource consumption, this novel approach aims to address the limitations of existing methods and provide a more practical solution for real-world applications.

In conclusion, the motivation for this research lies in the identification and mitigation of existing shortcomings in traditional and learning-based segmentation methods for spray macroscopic parameter determination. The proposed lightweight neural network segmentation method holds significant promise for the advancement of this field, as it offers a more streamlined and

resource-friendly alternative without compromising accuracy and performance.

4.2 Data augmentation

The dataset, explained in Chapter 3, employed in this study was partitioned into three distinct subsets, each corresponding to a specific phase of the neural network learning process: training, testing, and validation. The dataset was divided in a 60:20:20 ratio, with 60% allocated for training the models, 20% dedicated to validation, and the remaining 20% reserved for testing the models. The training and validation subsets underwent data augmentation, while the test subset was left unaltered.

Data augmentation is a widely recognized technique for expanding the pool of labeled samples, thereby enhancing the performance and generalization capabilities of deep learning models [56]. This process entails applying a series of transformations to both the data images and their corresponding labeled segmentation maps. In the present experiment, rotation and scaling transformations were implemented. A random integer within the interval [0°, 360°] was chosen, and the image was subsequently rotated in a clockwise direction. Regarding scaling, a random number within the range [0.1, 1.9] was selected, and the image was accordingly scaled. An example batch is visualized in Fig. 4.1.

Employing data augmentation significantly increased the number of images utilized for training the models. The initial count of 120 images, which constituted 60% of the entire dataset, was augmented to 120 $\times$ the number of epochs. This substantial expansion was achieved by applying the aforementioned rotation and scaling methods to each image within every batch during every epoch. Consequently, the models benefited from a more diverse and extensive training dataset, promoting their overall performance and adaptability to new data.

4.3 Evaluation metrics

The evaluation of performance metrics plays a crucial role in assessing the efficacy of both traditional computational methodologies and advanced deep neural network architectures. In the present study, the results were examined and compared utilizing the Dice Coefficient, a well-established metric originally introduced by Dice in 1948 [57]. The Dice Coefficient offers a quantitative measure of the degree of overlap between the ground truth and the predicted output.

Given two distinct sets A and B, the Dice Coefficient is formally defined as follows:

$$Dice = \frac{2|A \cap B|}{|A| + |B|} \tag{4.1}$$

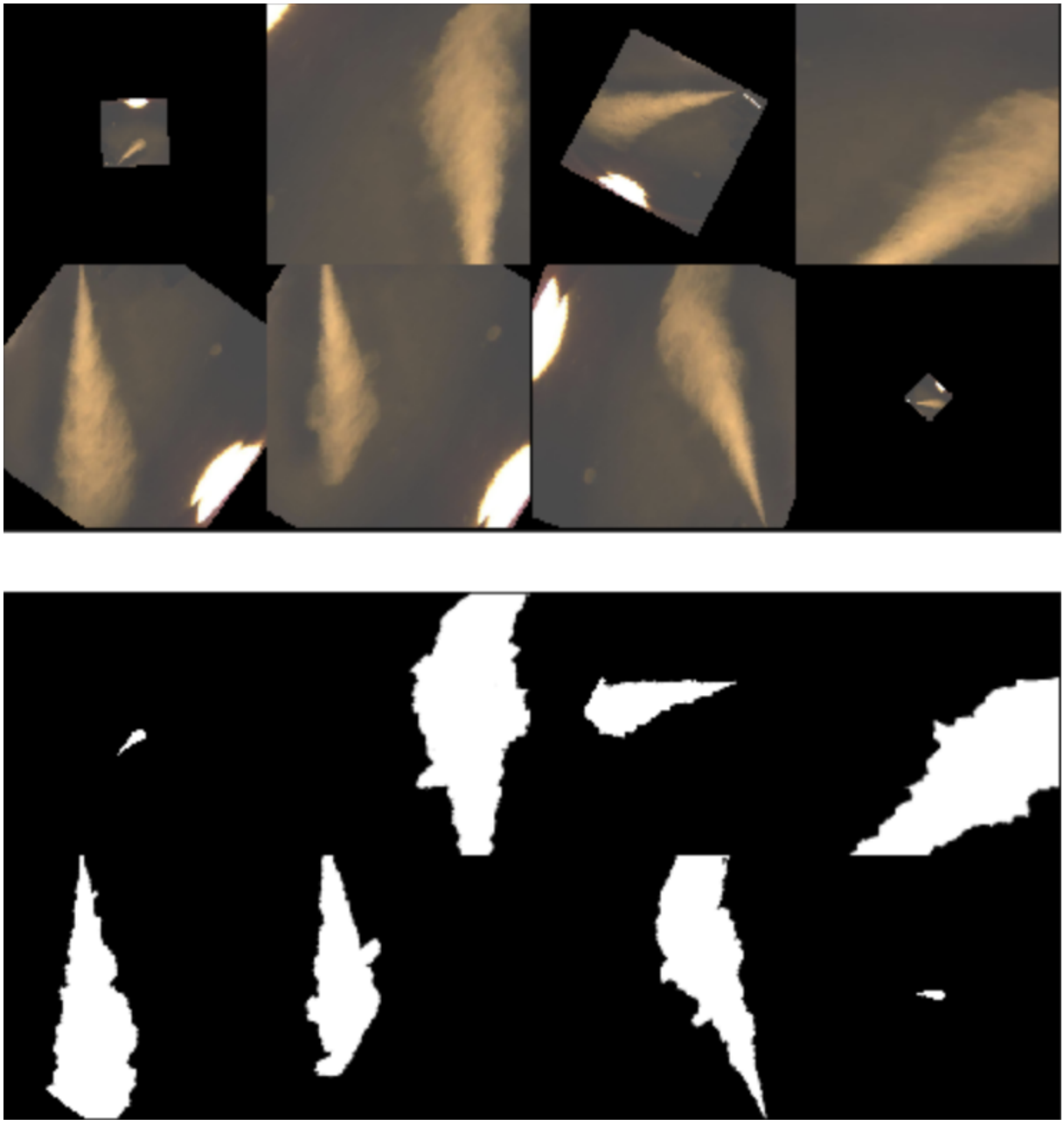

Figure 4.1: Example of augmented images (upper part) and their labels (lower part)

In this equation, $|A|$ and $|B|$ represent the number of elements contained in each respective set.

It is important to note that the Dice Coefficient shares a close resemblance with the classical Intersection Over Union (IoU) metric. In particular, when applied to binary maps, the Dice Coefficient effectively functions as an F1 score, as illustrated by the following equation:

$$Dice = \frac{2TP}{2TP + FP + FN} = F1 \tag{4.2}$$

In this context, TP, FP, and FN denote true positives, false positives, and false negatives, respectively. This equivalence further highlights the versatility and robustness of the Dice Coefficient as a valuable metric for evaluating the performance of diverse computational models and deep learning paradigms.

4.4 Proposed method

As depicted in Fig. 4.2, the proposed model, called Min U-Net, is composed of an encoder and a decoder, both meticulously designed to process input data and generate outputs effectively. The encoder component receives an RGB image as input, which is subsequently processed through a *DoubleConv* block. This particular block comprises two convolutional layers, each featuring four 3×3 kernels, accompanied by batch normalization and ReLU activation functions. It is important to note that all convolutional layers within the *DoubleConv* block possess a kernel size of 3×3.

Following the *DoubleConv* block, the output is directed through two sequential *Down* blocks. The first *Down* block incorporates a 2×2 max-pooling layer with a stride size of two, as well as a *DoubleConv* block containing eight kernels in both convolutional layers. Similarly, the second *Down* block features a *DoubleConv* block with eight convolution kernels.

Subsequent to the last *Down* layer, the decoder component commences. This section is comprised of two *Up* blocks, each of which includes bilinear upsampling with a scale factor of 2.0 and a *DoubleConv* block. The input for each *Up* block consists of the output from the preceding layer, concatenated with the output of the corresponding *Down* block at the same depth. In the first *Up* block, the initial convolutional layer contains eight kernels, while the subsequent convolution features four kernels. In the second *Up* block, both convolutional layers consist of four kernels.

Ultimately, the feature map is processed through a singular 1×1 convolutional layer to generate a binary segmentation map. A comprehensive visual representation of the model architecture can be found in Fig. 4.2.

Upon the completion of image segmentation, the macroscopic parameters of the spray are ascertained from the segmented image, as per the methodology delineated in this study. To achieve the appropriate right-hand orientation, the image is rotated using Principal Component

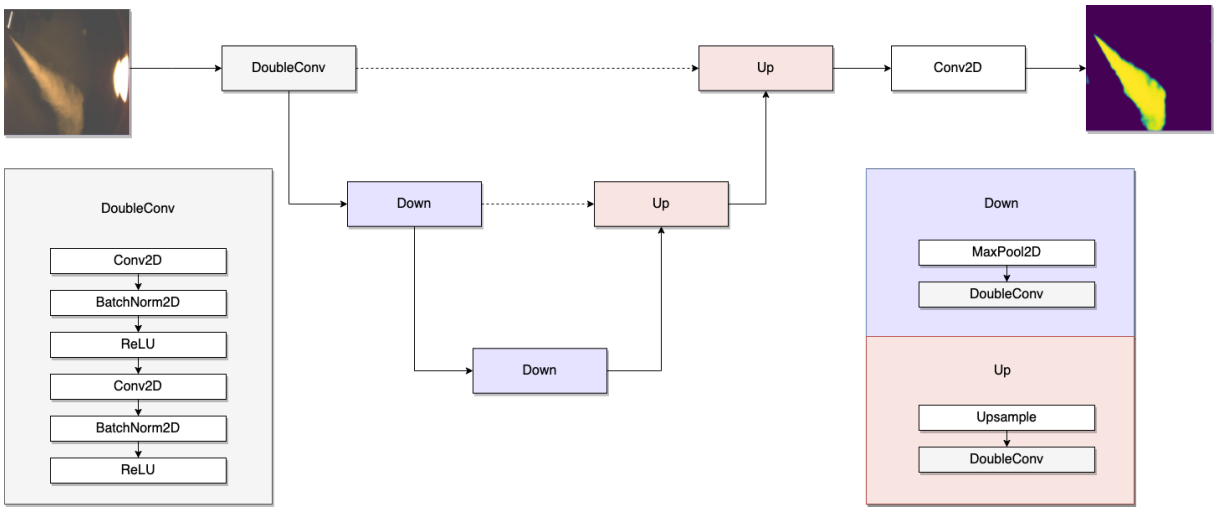

Figure 4.2: Min U-Net architecture for spray segmentation. The model receives an RGB image as input and produces a segmentation map as output. It features a single *DoubleConv* block, a pair of *Down* and *Up* blocks, and one convolutional layer, arranged in a characteristic U-Net pattern. Solid lines illustrate the data flow, while dashed lines denote skip connections within the model.

Analysis (PCA), subsequently allowing for the detection of spray edges. By employing the spray edges in conjunction with a line fitting algorithm, essential parameters such as the cone angle, spray penetration length, and spray area are computed. A comprehensive explanation regarding the determination of these macroscopic spray parameters can be found in Sec. 3.3. The proposed method's pseudo-code is provided in Algorithm 1.

Algorithm 1 Estimation of Macroscopic Spray Parameters

Require: Input image

$model \leftarrow MinUNet()$
$input\_image \leftarrow PreprocessImage(input\_image)$
$output\_image \leftarrow model(input\_image)$
$orientation\_angle \leftarrow PCA(output\_image)$
$rotated\_output\_image \leftarrow RotateImage(output\_image, orientation\_angle)$
$macroscopic\_parameters \leftarrow DetermineParameters(rotated\_output\_image)$
return $macroscopic\_parameters$

In summary, an instance of the Min U-Net network is initially generated. Subsequently, the image undergoes preprocessing to facilitate segmentation by the Min U-Net. Once the segmentation process is complete, the orientation angle is derived using PCA, and the image is rotated accordingly. Finally, the rotated segmented image serves as the basis for determining the macroscopic spray parameters, effectively concluding the analysis.

4.5 Ablation study

In order to determine the most appropriate model architecture for addressing the problem of spray segmentation as outlined in Section 4.4, an ablation study was conducted. This study comprised two procedures, namely, the investigation of the U-Net depth level and the number of kernels in each convolutional layer to arrive at the proposed method. An additional approach to minimize the number of parameters and complexity involved the utilization of bilinear up-sampling in the *Up* blocks as opposed to employing transposed convolution, which subsequently reduced the number of channels in the feature map.

The fundamental objective of this research is to streamline the U-Net architecture while preserving its accuracy. This simplification is achieved by eliminating layers and diminishing the number of kernels in the convolutional layers. In the original U-Net architecture, the number of kernels in the encoder portion doubles at every level, while it is halved at every level in the decoder portion. The Min U-Net follows this procedure but employs a significantly smaller number of kernels in order to minimize complexity. By decreasing the number of kernels, the model's complexity is also reduced, further contributing to the simplification of the spray segmentation model. The number of kernels in the convolutional layers is also minimized. Algorithm 2 illustrates the pseudo-code for the ablation study.

Algorithm 2 Number of kernels in blocks

Require: $kernel \geq 1$
Require: $depth \geq 1$
Set both convolutional layers in *DoubleConv* to number of kernels *kernel*
$layers \leftarrow depth - 1$
for $i \leftarrow 0, layers$ **do**
    if $i = layers$ **then**
        $a \leftarrow kernel * 2^i$
    else
        $a \leftarrow kernel * 2^{(i+1)}$
    end if
    Set both convolutional layers in $Down_i$ to number of kernels a
end for
for $i \leftarrow layers, 0$ **do**
    if $i = 0$ **then**
        $b \leftarrow kernel * 2^i$
    else
        $b \leftarrow kernel * 2^{(i-1)}$
    end if
    $a \leftarrow kernel * 2^i$
    Set first convolutional layer in Up_i to number of kernels $\frac{a}{2}$ and second convolutional layer to b
end for
Set final convolutional layer to 1 kernel

As per the pseudo-code, the kernel and depth number must be equal to or greater than one. Once the starting kernel number and depth are established, both convolutional layers in the *DoubleConv* are set to the given number of kernels. For each layer in the encoder portion, both convolutional layers in the *Down* block are set to double the number of kernels of the preceding layer, except for the final layer. In the last layer, the number of kernels remains identical to the layer before. In the decoder portion, the process begins with the deepest layer, where the first convolution in the *Up* block is set to the same kernel number as the second convolutional layer from the previous block. The second convolution in the block is set to half the number of the first convolution from the same block. In the final *Up* block, both convolutions are assigned the same number of kernels as the second convolution from the preceding block. In the concluding single convolutional layer, both kernel numbers are set to one. The model's complexity is directly influenced by the number of kernels and the depth level.

Table 4.1 displays the number of kernels in the *DoubleConv* block at depth 1. Given the established pattern of doubling kernel numbers in the encoder and halving them in the decoder, it is possible to deduce the number of kernels in all other layers. For instance, assuming a depth of 4 and a starting kernel number of 8, both convolutional layers in the *DoubleConv* block contain 8 kernels, whereas both convolutional layers in the first *Down* block comprise 16 kernels. Furthermore, both convolutional layers in the second and third *Down* blocks have 32 kernels each. In the first *Up* block, the first convolutional layer has 32 kernels while the second one has 16 kernels. The second *Up* block consists of 16 and 8 kernels, and finally, both convolutional layers in the third *Up* block possess 8 kernels each.

Table 4.2 demonstrates the impact of increasing depth and kernel numbers on the total number of parameters in a model. The smallest model, with a depth of 1 and a single starting kernel, consists of merely 44 parameters. In stark contrast, the largest model encompasses 4.32 million parameters, rendering it comparable to the baseline models discussed in Section 4.6.1. Models with a greater number of parameters tend to achieve higher levels of accuracy. However, as evidenced by Tables 4.1 and 4.2, there exists a point of diminishing returns in terms of complexity for the problem of spray segmentation.

Four distinct models, each achieving a mean Dice coefficient of 0.95 and characterized by a reduced number of parameters, were compared to identify the most optimal among them. The evaluation metrics employed in the comparison included the mean, median, best 25%, worst 25%, and trimean values. The results of this comparison are presented in Table 4.3. As mentioned earlier, all models attain a mean of 0.95, a best 25% of 0.96, and a trimean of 0.95. The model with a depth of 3 and a kernel number of 4 exhibits the highest Dice coefficient median of 0.96 and the best performance on the worst 25% metric, scoring 0.94, while the other models achieve a median of 0.95 and the worst 25% of 0.93. The model with a depth of 4 and a kernel number of 4 possesses the highest number of parameters, totaling 17 thousand, whereas

Table 4.1: Comparison of Dice coefficients for proposed models, organized by depth in columns and initial kernel number in rows

	1	2	3	4	5
1	0.87	0.90	0.78	0.92	0.91
2	0.90	0.92	0.92	0.95	0.95
4	0.91	0.93	**0.95**	0.95	0.95
8	0.91	0.95	0.94	0.95	0.95
16	0.90	0.94	0.95	0.95	0.95
32	0.94	0.94	0.95	0.95	0.95

Table 4.2: Comparison of the number of parameters for proposed models, organized by depth in columns and initial kernel number in rows

	1	2	3	4	5
1	44	0.1K	0.3K	1.2K	4.4K
2	0.1K	0.3K	1.1K	4.4K	17.3K
4	0.2K	1.0K	**4.3K**	17.2K	68.3K
8	0.8K	3.8K	16.7K	67.8K	271.5K
16	2.8K	14.6K	65.7K	269.3K	1.08M
32	10.3K	56.8K	260.4K	1.07M	4.32M

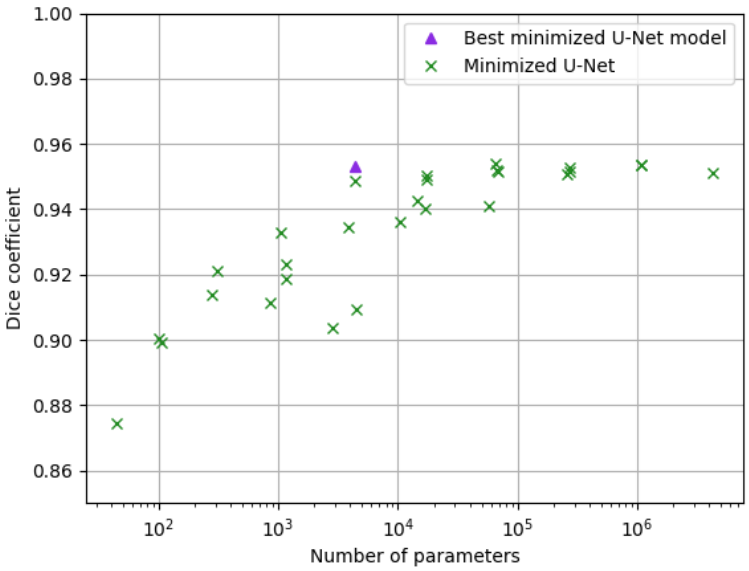

Figure 4.3: Illustration of the graphical comparison of all minimized U-Net variations with respect to Dice coefficients and the number of parameters

the remaining models exhibit similar parameter counts. Consequently, the model with a depth of 3 and a kernel number of 4 was chosen for further analysis in this study and is proposed in Section 4.4.

Table 4.3: Comparision of a few better minimized U-Net models

	Depth level - Kernel number			
	3 - 4	4 - 2	2 - 8	4 - 4
Number of parameters	4.3K	4.4K	3.8K	17K
Mean	0.95	0.95	0.95	0.95
Median	**0.96**	0.95	0.95	0.95
Best 25%	0.96	0.96	0.96	0.96
Worst 25%	**0.94**	0.93	0.93	0.93
Trimean	0.95	0.95	0.95	0.95

4.6 Experimental results

4.6.1 Segmentation results

A series of experiments were conducted to determine the most suitable minimized model for spray segmentation. As demonstrated in Section 4.5, highly complex baseline models appear to be excessive for addressing this particular problem. Traditional methods, despite achieving reasonable results, lack the requisite complexity to attain the accuracy level of baseline models. By optimally minimizing the U-Net architecture, complexity is reduced without compromising accuracy, rendering Min U-Net models the ideal solution for spray segmentation tasks.

In this section, a comprehensive comparison of results obtained from traditional methods, baseline models, and proposed models is presented. Also the methods are explained in detail.

The learning-based models were trained using augmented images from Section 4.2 and evaluated on the test set of the dataset, which consisted of images rescaled to a size of 256×256. All proposed models underwent training for 100 epochs, employing the Adam optimizer with an initial learning rate of 0.00001, while the baseline models utilized an initial learning rate of 0.0001. Notably, the baseline models were pretrained on the well-known ImageNet dataset [58]. Moreover, to address the challenges posed by training with an unbalanced dataset, a cyclic learning rate with an upper rate of 0.1 was implemented [59].

Traditional methods

Prior to the advent of deep learning techniques, the most prevalent approaches for addressing the segmentation problem were classified as traditional methods. These methods encompass K-Means segmentation, thresholding, the Otsu method, max entropy segmentation, and others [60, 61, 62, 63, 64].

In this subsection, the results of image segmentation employing three well-established traditional methods—thresholding, the Otsu method, and max entropy segmentation—are presented [9].

Simple thresholding is a fundamental image segmentation technique that assigns each pixel $I_{i,j}$ to either the foreground or background class based on a predefined threshold value T. This method is widely used in various applications due to its simplicity and computational efficiency. The primary objective of simple thresholding is to separate an image into meaningful regions based on pixel intensity. To apply simple thresholding, the input image is first converted to grayscale, which reduces the complexity of the segmentation process by representing the image in a single intensity channel. Next, a threshold value is selected, which can be determined manually or automatically using various approaches, such as analyzing the image histogram or using an iterative method. The threshold value serves as a criterion for dividing the image into two distinct classes: the foreground and the background. Once the threshold value is established, the segmentation process begins by comparing each pixel's intensity value to the threshold. If a pixel's intensity is greater than or equal to the threshold value ($I_{i,j} < T$), it is assigned to the foreground class (commonly represented by white); otherwise, it is assigned to the background class (commonly represented by black). The result is a binary image, where each pixel belongs to either the foreground or background class.

Otsu thresholding is an influential global thresholding technique in image segmentation, specifically designed for discriminating objects from the background. Proposed by Nobuyuki Otsu in 1979 [8], the method is grounded in the principle of maximizing the between-class variance or minimizing the within-class variance. The technique assumes that the image to be segmented embodies two classes of pixels, foreground and background, and endeavors to identify the optimal threshold value to delineate these two classes.

To apply Otsu thresholding, the input image is initially converted to grayscale, followed by the calculation of its histogram. The histogram illustrates the distribution of pixel intensities, with the x-axis signifying intensity values and the y-axis depicting the frequency of each intensity value. Subsequently, the probability of each intensity level is computed by dividing the frequency of each intensity value by the total number of pixels in the image, resulting in the normalized histogram.

Cumulative sums for each intensity level are then calculated, providing the cumulative distribution function (CDF) of the intensity values. Following this, the cumulative mean for each

intensity level is computed to determine the expected value of intensity for both the foreground and background classes. The variance between these two classes, or the between-class variance, is calculated for each possible threshold value.

The Otsu thresholding technique identifies the optimal threshold value as the one that maximizes the between-class variance or equivalently minimizes the within-class variance. Once the optimal threshold value T^* is determined, the image is segmented into foreground and background classes by assigning pixel values above the threshold to the foreground class and pixel values below the threshold to the background class, or in this case the background and the spray.

To find the optimal threshold value T^* by maximizing the variance between the classes as explained in Eg. 4.6.1.

$$T^* = \underset{0 \leq t < L}{\arg\max} \{\sigma_B^2(t)\} \tag{4.3}$$

where the variance between classes σ_B is defined as:

$$\sigma_B(t) = \omega_1(t)(\mu_1(t) - \mu_T)^2 + \omega_2(t)(\mu_2(t) - \mu_T)^2 \tag{4.4}$$

Max entropy segmentation, or maximum entropy thresholding, is an image segmentation technique that maximizes the entropy of the resulting foreground and background classes [61, 62]. Entropy represents the information content or randomness within an image. This method is particularly useful for complex and noisy images, as it finds an optimal threshold value that best separates the two classes.

The input image is first converted to grayscale and its histogram is calculated. The normalized histogram is then computed by dividing the frequency of each intensity value by the total number of pixels. Entropy for the foreground and background classes is calculated for each possible threshold value, and the combined entropy is determined as the sum of the individual entropies.

Max entropy segmentation threshold T that results in the largest entropy is calculated as:

$$T = \underset{0 \leq s \leq L-1; 0 \leq t \leq L-1}{\arg\max} H(t) \tag{4.5}$$

where $H(t)$ is the total entropy that can be calculated as:

$$H(t) = \log p_1(t) p_2(t) + \frac{h_1(t)}{p_1(t)} + \frac{h_2(t)}{p_2(t)} \tag{4.6}$$

The traditional segmentation methods were evaluated on both original and cropped images. Cropped images are those that have been processed to include only the spray region, effectively

eliminating background noise, such as the lamp on the right side and the small lamp on the left side where the spray originates mentioned previously in Sec. 3.2. As illustrated in Table 4.4, applying these methods to cropped images results in a significant improvement in segmentation performance compared to the original images.

Among the traditional methods, simple thresholding demonstrated the lowest accuracy, achieving a Dice coefficient of 0.46 for original images and 0.77 for cropped images. The Otsu method exhibited a substantial improvement of 57% when applied to cropped images, increasing the Dice coefficient from 0.52 to 0.82. The most effective traditional method was max entropy segmentation, which achieved a mean Dice coefficient of 0.79 for original images and a slight increase to 0.85 for cropped images.

Figure 4.4: Example of original and cropped image

While the results obtained from cropped images represent a considerable enhancement compared to the original images, it is important to note that cropped images represent an idealized, noiseless scenario specific to this particular dataset. The applicability of the cropping technique to other datasets is limited, as it would likely require customized and potentially complex adjustments for each new dataset.

Table 4.4: Dice coefficient comparison of traditional methods

	Without crop	With crop
Tresholding	0.46	0.77
Otsu	0.52	0.82
Max Entropy	0.79	0.85
Min U-Net	**0.95**	**0.96**

In contrast to traditional methods, the proposed Min U-Net model demonstrates superior performance, achieving a Dice coefficient of 0.95 for original images and 0.96 for cropped images, despite being trained on original images. This superior performance underscores the

advantages of the Min U-Net model in addressing the spray segmentation problem over traditional methods.

Another area where the Min U-Net demonstrates superior performance is in handling artificial noise introduced to the input image. As illustrated in Figure 4.5, traditional methods experience a substantial decline in accuracy when confronted with the presence of white squares and Gaussian noise. These findings underscore the robustness and resilience of the Min U-Net model in maintaining its effectiveness despite the presence of such noise, further highlighting its suitability for various applications in the domain of image segmentation.

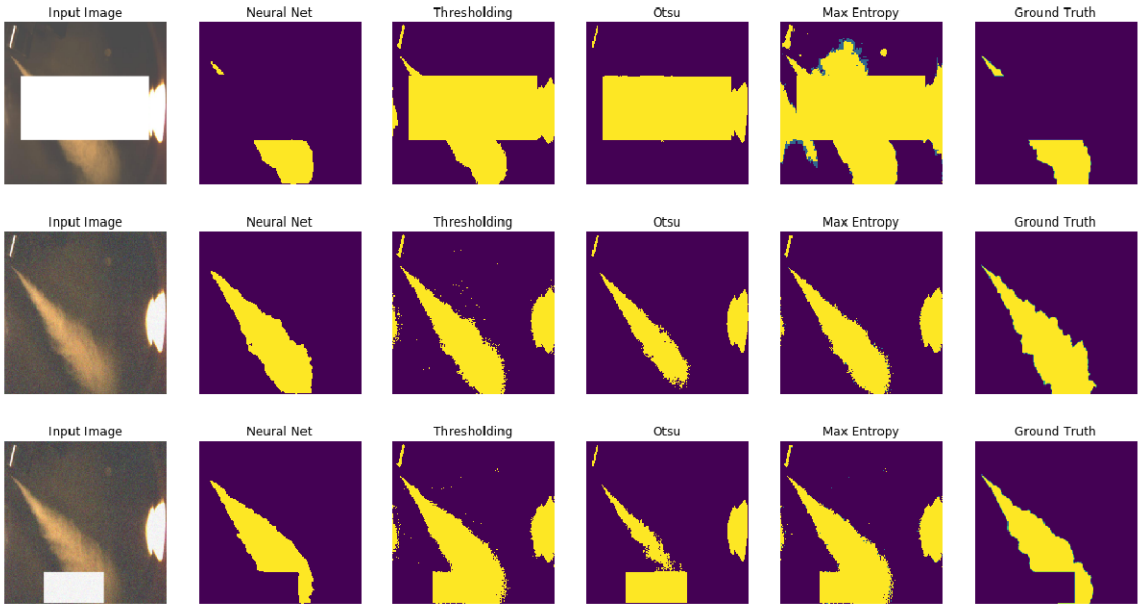

Figure 4.5: A visual comparison of model outputs for traditional methods. The first row displays the input image containing a random white square, the second row presents the image with added Gaussian noise, and the third row combines both types of noise (white square and Gaussian noise)

Baseline comparison

Deep learning methods have been employed to address the challenge of situation-specific cropping in image segmentation tasks. As evidenced in numerous applications, deep learning has emerged as a powerful technique for semantic segmentation, offering the potential to surpass the performance of traditional methods in terms of accuracy, speed, and full automation. The core strength of deep learning lies in its ability to extract and learn relevant features from the input data, a capability not inherent in traditional methods. In contrast to traditional approaches, deep learning methods possess an innate capacity for knowledge representation and generalization, enabling them to adapt and excel in a wide array of segmentation tasks.

Four state-of-the-art models of architecture with six different backbones are trained and evaluated. Architectures that were utilized are Feature Pyramid Network (FPN), Linknet, Pyra-

mid Scene Parsing Network (PSPNet), and U-Net.

Feature Pyramid Network (FPN) is a deep learning architecture that constructs a multi-scale feature pyramid by combining high-level semantic information with low-level detailed information. It efficiently leverages a top-down pathway with lateral connections to merge coarse but semantically rich features from higher layers and fine-grained spatial information from lower layers. FPN is widely used in object detection and segmentation tasks, as it enables the detection of objects at various scales and improves the overall performance of the network [65]. LinkNet is a deep learning architecture designed for efficient semantic segmentation tasks. It employs an encoder-decoder structure with skip connections between corresponding layers in the encoder and decoder, allowing the network to reuse features from the encoding path and retain spatial information during the decoding process. This approach enables LinkNet to generate high-quality segmentation outputs while maintaining a relatively lightweight model architecture [66]. PSPNet, or Pyramid Scene Parsing Network, is a deep learning architecture developed for semantic segmentation tasks. It employs a pyramid pooling module that combines global and local context information at multiple scales, enabling the model to capture both fine-grained and coarse-level details. By integrating this module with a backbone network, such as a deep convolutional neural network, PSPNet effectively handles varying object sizes and complex scenes, resulting in improved segmentation performance across a wide range of applications [51]. U-Net is the most famous and frequently used type of convolutional neural network. Its original use was for biomedical image segmentation, but it has been used for many applications since its publication [67]. U-Net has two main parts, the encoder, and the decoder. The encoder is used for downsampling while the decoder is used for upsampling. The input is a grayscale image processed through blocks of two 3×3 convolutional layers with ReLu activation function. After the convolutional layers, there is a 2×2 max-pooling layer with a stride size of 2 for downsampling. The decoder part consists of blocks of the upsampling of the feature map with convolution filters of size 2×2. The feature map is then concatenated with the cropped feature map from the encoder part and passed to two 3×3 convolution layers with ReLu activation function. The final layer is a 1×1 convolutional layer mapping the feature maps to a given number of classes

Six backbones that were applied to the models are:

- DenseNet121: DenseNet (Densely Connected Convolutional Networks) is a deep learning architecture that focuses on improving information flow and gradient propagation by connecting each layer to every other layer in a feed-forward fashion. DenseNet121 is a specific variant with 121 layers. This architecture leads to more efficient feature reuse and reduced number of parameters compared to traditional CNNs [58].
- DPN68: Dual Path Networks (DPNs) combine the strengths of ResNets and DenseNets by utilizing both residual and densely connected paths within the network. DPN68 is a

variant with 68 layers, and it leverages the benefits of both architectures to achieve better performance in image classification tasks [68].

- •EfficientNet-B0: EfficientNet is a family of neural network architectures that use a compound scaling method to optimize depth, width, and resolution of the network simultaneously. EfficientNet-B0 is the baseline model in this family, providing a balance between accuracy and computational complexity, making it suitable for a wide range of applications [69].
- •MobileNetV2: MobileNetV2 is a lightweight and efficient deep-learning architecture designed for mobile and embedded vision applications. It employs depthwise separable convolutions and inverted residual blocks with linear bottlenecks to reduce computational cost while maintaining high performance on image classification and object detection tasks [70].
- •ResNet34: ResNet (Residual Network) is a deep learning architecture that introduces skip connections or shortcuts between layers to combat the vanishing gradient problem in deep networks. ResNet34 is a variant with 34 layers, providing a balance between computational complexity and performance, making it suitable for various computer vision tasks [71].
- •VGG11: VGG11 is a deep convolutional neural network architecture, part of the Visual Geometry Group (VGG) family of models. It has 11 weight layers, including 8 convolutional layers and 3 fully connected layers. VGG11 is characterized by its simplicity and use of small 3x3 convolutional filters, allowing it to efficiently capture local context and spatial hierarchies in input images [72].

In Table 4.5, the mean dice coefficients are presented for various combinations of model architectures and backbones. The lowest dice score, 0.94, is attained when evaluating the PSPNet with an *efficientnet-b0* backbone. All other baseline models achieve a minimum dice score of 0.95. It is important to note that these baseline models exhibit a high learning capacity due to their substantial number of parameters, as illustrated in Table 4.6. The least complex baseline model, PSPNet with a *mobilenet_v2* backbone, comprises 2.3 million parameters. In contrast, the most complex model, U-Net with a *resnet34* backbone, contains 24.4 million parameters. Consequently, these baseline models exhibit an exceedingly high level of complexity for the task of spray segmentation, which may not be necessary for achieving optimal performance in this specific problem domain.

The Min U-Net demonstrates results comparable to those of the baseline models, while possessing more than 500 times fewer parameters than the least complex baseline model and over 5600 times fewer parameters than the most complex baseline model. This significant reduction in model complexity renders the Min U-Net more hardware-friendly and computationally efficient compared to the larger, more complex baseline models. Figure 4.6 illustrates the stark

Table 4.5: Dice coefficient comparison of baseline models

	FPN	Linknet	PSPNet	U-Net
densenet121	0.96	0.96	0.96	0.96
dpn68	0.95	0.95	0.95	0.96
efficientnet-b0	0.96	0.95	0.94	0.96
mobilenet_v2	0.95	0.96	0.95	0.96
resnet34	0.95	0.96	0.95	0.96
vgg11	0.95	0.96	0.96	0.95
Min U-Net	-	-	-	0.95

Table 4.6: Number of parameters comparison of baseline models

	FPN	Linknet	PSPNet	U-Net
densenet121	9.3M	10.4M	7.7M	13.6M
dpn68	13.9M	13.6M	12.2M	17.0M
efficientnet-b0	5.8M	4.2M	4.1M	6.3M
mobilenet_v2	4.2M	4.3M	2.3M	6.6M
resnet34	23.2M	21.8M	21.4M	24.4M
vgg11	11.3M	10.5M	10.0M	18.3M
Min U-Net	-	-	-	4.3K

difference in the number of parameters between the Min U-Net and the baseline models, with the x-axis presented on a logarithmic scale. This comparison highlights the potential advantages of utilizing the Min U-Net architecture for spray segmentation tasks, particularly in resource-constrained environments or when computational efficiency is a priority.

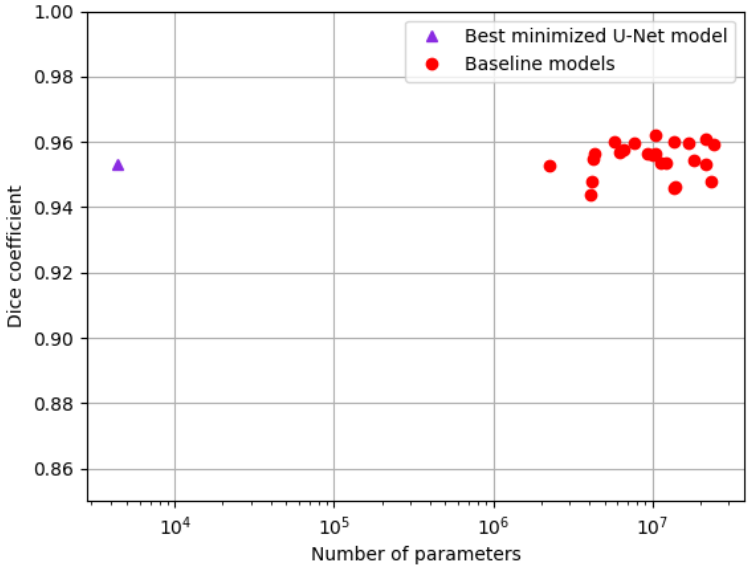

Figure 4.6: Comparison Min U-Net with baseline models when looking at dice coefficient and number of parameters

Another noteworthy benefit of the Min U-Net surpassing the performance of the baseline models lies in its reduced inference time. Owing to its streamlined architecture and fewer parameters, the Min U-Net requires only 11.94 ms to process an image, which is over two times faster than the quickest baseline model, as illustrated in Fig. 4.7. This remarkable efficiency in processing speed further reinforces the Min U-Net's suitability for spray segmentation tasks, particularly in real-time applications or scenarios where rapid response times are crucial.

4.6.2 Macroscopic spray parameters results

In order to compute spray macroparameters, noise must be removed from the image, as detailed in Section 3.3. The most prominent contour is identified, and all other components in the segmented image are eliminated, with pixel values set to zero. This background removal process significantly impacts traditional methods, which struggle to differentiate spray from noise, particularly during the initial phases of spray injections, as evidenced in Figs. 4.8 to 4.10. Traditional methods tend to segment the lamp situated on the right side of the image, which becomes the most prominent contour if the spray is inadequately segmented, especially when employing the Otsu method. It is notable that the Otsu method yields the smallest area among all methods, which subsequently affects the determination of cone angle and penetration.

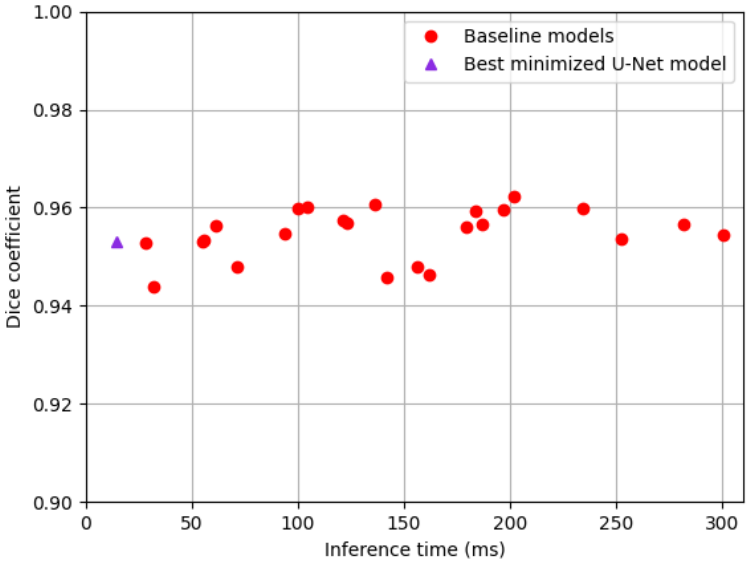

Figure 4.7: Speed comparison of Min U-Net baseline models relative to dice coefficient

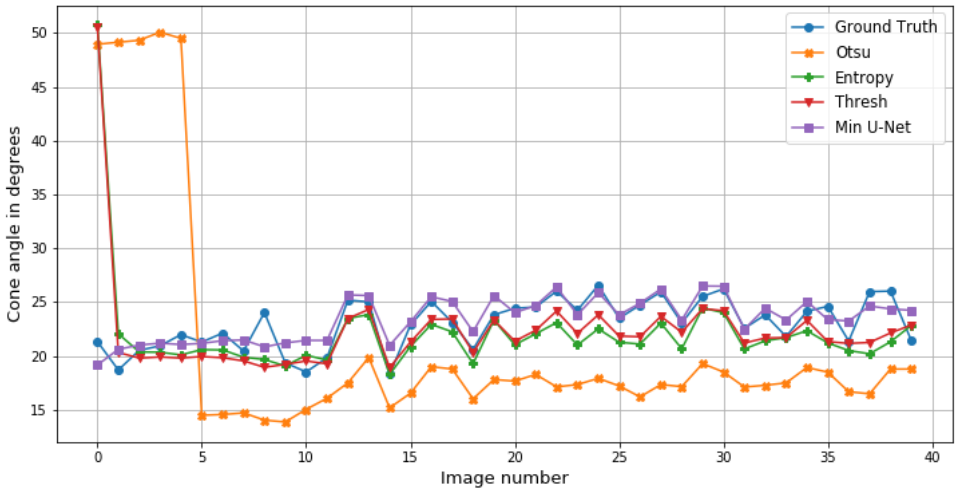

Figure 4.8: Comparision of cone angles with different segmentations methods

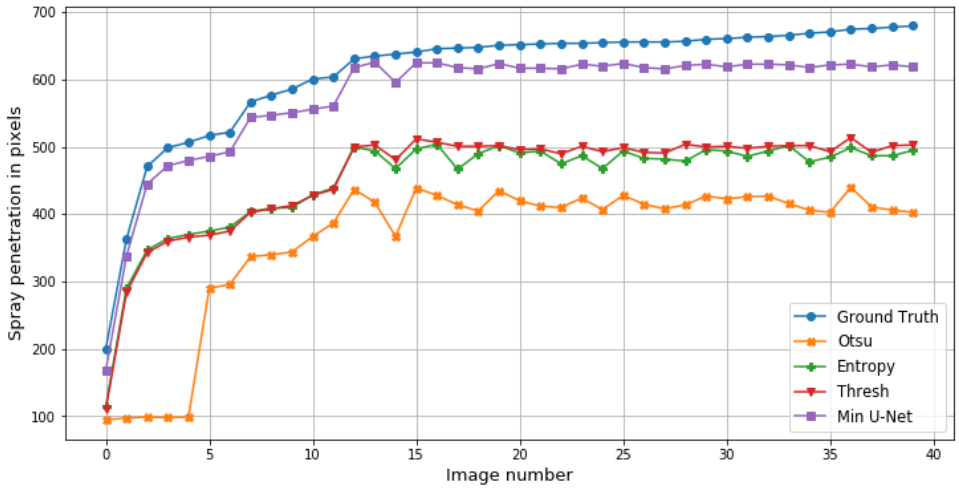

Figure 4.9: Comparision of spray penetration with different segmentations methods

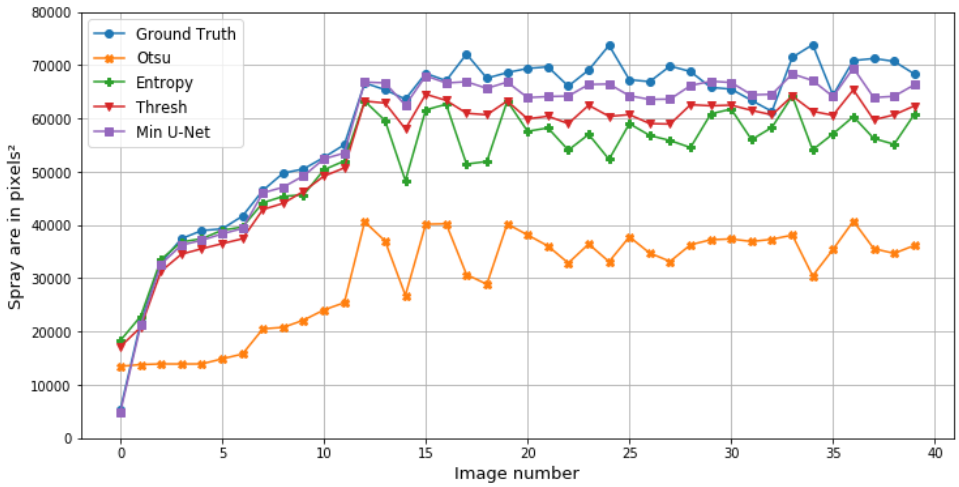

Figure 4.10: Comparision of spray area with different segmentations methods

Table 4.7 compares the results of traditional methods with those of the Min U-Net. The table presents the final outcomes, which demonstrate the effectiveness of the proposed approach. The mean absolute error (MAE) is computed for cone angles, while the mean relative error (MRE) is calculated for spray penetration and area. As illustrated in the table, Min U-Net achieves substantially smaller errors. Min U-Net exhibits a mean absolute error of 1.08 degrees for the cone angle, and mean relative errors of 5.95% and 4.05% for spray penetration and spray area, respectively.

In terms of cone angle, Min U-Net outperforms simple thresholding, which is the best performing traditional method for all macroscopic parameters, by more than $2\times$. For spray penetration, Min U-Net surpasses simple thresholding by over $4\times$, and for spray area, by over $3.5\times$. The Otsu method yields the largest errors due to the aforementioned issues, while the Max Entropy segmentation generates results similar to those of simple thresholding.

Table 4.7: Spray macroparameters mean absolute errors comparison of traditional methods and Min U-Net

	Cone angle MAE	Spray penetration MRE (%)	Spray area MRE (%)
Thresholding	2.36	25.55	14.36
Otsu	9.05	41.84	52.51
Max Entropy	2.68	26.47	18.05
Min U-Net	**1.08**	**5.95**	**4.05**

Chapter 5

Regression-based spray sequence macroscopic parameters measurement

5.1 Motivation

The primary motivation for developing the proposed neural network regression method, which employs image sequences rather than single images, arises from the strong correlation observed among images captured during a single spray injection event. To the best of our knowledge, no existing deep learning-based methods leverage this intrinsic relationship among spray sequence images to accurately determine spray macroscopic parameters.

The secondary motivation, as outlined in Sec. 4.1, is to enhance engine efficiency and reduce pollution emissions from internal combustion engines. A majority of the existing literature methods focus on the use of a single image for determining spray macroscopic parameters, thereby neglecting the valuable information contained in the related and correlated images captured during the same event.

Given that the images are acquired using a high-speed camera, as detailed in Sec. 3.1, the proposed method aims to harness the information embedded in previous images and their corresponding labels to achieve a more accurate parameter determination compared to the single-image-based approaches. By exploiting the temporal dependencies among sequential images, this novel methodology seeks to provide a more comprehensive understanding of the spray injection process, ultimately leading to improved parameter estimation.

In summary, the motivation behind the proposed neural network regression method lies in capitalizing on the wealth of information available in spray sequence images to advance the state-of-the-art in determining spray macroscopic parameters.

5.2 Data

5.2.1 Preprocessing

In order to achieve a uniform appearance across the analyzed images, a series of preprocessing steps were systematically applied. Initially, the images were transformed from their original RGB color space into grayscale values, simplifying the subsequent processing stages. Following this transformation, each image was element-wise multiplied with its corresponding segmentation mask, effectively eliminating the background and isolating the region of interest.

Subsequently, the Principal Component Analysis (PCA) technique, as described in Section 3.3, was employed to obtain the spray orientation. Upon determining the orientation angle, the image was rotated accordingly to establish a consistent right-hand orientation across all samples. This rotation ensured uniformity in the dataset, facilitating further analysis and comparison.

To accurately position the spray's starting point in the images, the x and y coordinates were set to $x_{image} = 0$ and $y_{image} = \frac{image\_height}{2}$, respectively. This was achieved using the Affine transformation, as outlined in Equation 5.2.1, effectively centering the spray within the image frame.

$$T = M \times [x \quad y]^T \tag{5.1}$$

The spray start or the tip of the spray is defined with two points x_{start} and y_{start}, and the matrix M is defined as $M = \begin{bmatrix} 1 & 0 & -x_{start} \\ 0 & 1 & -(y_{start} - \frac{image\_height}{2}) \end{bmatrix}$.

A comprehensive illustration of the entire preprocessing pipeline, detailing the sequential steps from the initial RGB image to the final preprocessed image, can be found in Figure 5.1. This systematic approach ensures a consistent, uniform appearance of the images, thereby enhancing the reliability and validity of the subsequent analysis conducted in the context.

5.2.2 Data augmentation

Data augmentation serves as an effective technique to increase the size of training datasets, consequently improving the generalization capabilities of deep learning-based models [73]. Additionally, augmentation can expand the range of values represented in the training labels, thereby further enhancing the model's robustness and performance.

In the context of the present study, the cone angle labels exhibit a small standard deviation of merely 1.74 degrees, spanning a range from 17.25 to 26.03 degrees. The limited deviation in-

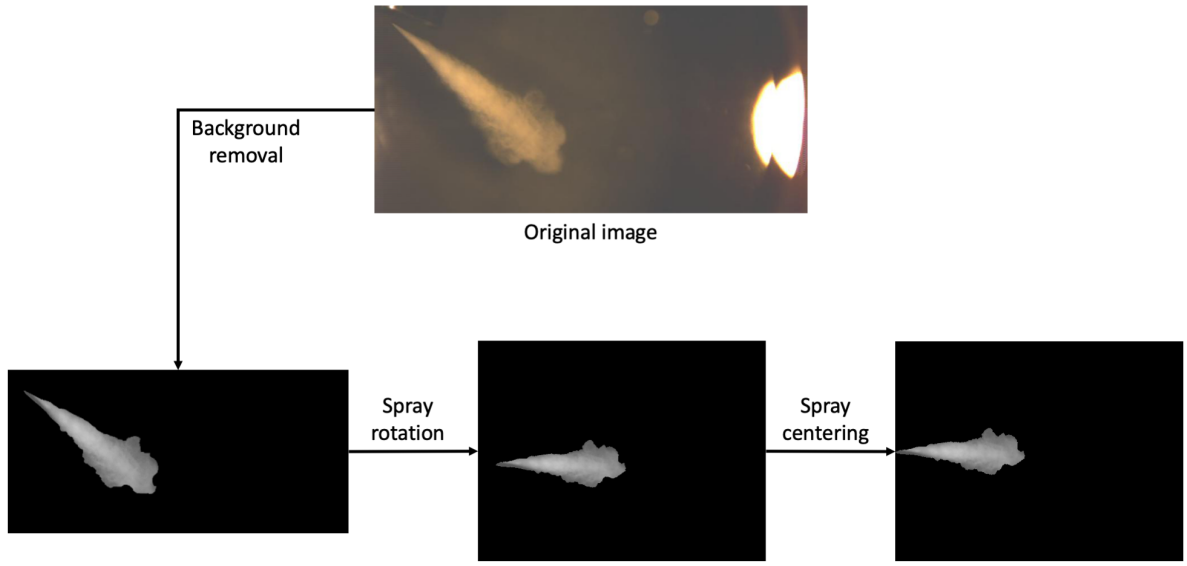

Figure 5.1: Image preprocessing pipeline visualization

creases the likelihood of overfitting and necessitates a broader range of cone angles. To address this issue, data augmentation was employed. Given the nature of the image preprocessing, the images can be readily manipulated along the y-axis, either by stretching or compressing them. Stretching the image and subsequently reshaping it results in an increased cone angle, whereas compressing the image leads to a decreased cone angle.

To achieve the compression of the image, and consequently the cone angle, a random number was selected from the range $[1, 1500]$. The image was then padded at the top and bottom with the chosen number of pixels. After padding, the image was resized back to its original height, effectively generating the appearance of a compressed spray and a reduced spray angle.

To attain the widening of the image and the corresponding cone angle, a random number N was selected from the range $[1, 2000]$. Subsequently, $\frac{N}{2}$ pixels were removed from both the top and bottom of the image by cropping. The image was then resized to its original dimensions, successfully achieving the effect of an expanded image and an increased spray angle. By employing these data augmentation techniques, the range of cone angles in the training dataset is effectively broadened, mitigating the risk of overfitting and enhancing the model's ability to generalize across various spray scenarios.

Following the implementation of the augmentation techniques, a notable increase in the standard deviation of cone angle labels was observed, rising to 12.91. Concurrently, the range of cone angles expanded, with the maximum angle reaching 58.52 degrees and the minimum angle registering at 5.09 degrees. Despite these variations, the mean and median values remained constant, as detailed in Table 5.1. The augmentation process effectively increased the training

set size from an initial count of 156 images to a more robust collection of 3276 images, thereby enhancing the representativeness of the dataset.

Table 5.1: Original and augmented spray cone angle label metrics

	Mean	Median	Std. deviation	Max	Min
Cone angle (degrees)	21.41	21.53	1.74	26.03	17.25
Augmented cone angle (degrees)	23.03	21.28	12.91	58.52	5.09

In addition to the aforementioned augmentations, a supplementary set of transformations was applied during the training process, specifically during the batching phase. These transformations encompassed rotation, flipping, and cropping operations. For the rotation operation, a random angle within the range $[0, 90]$ degrees was selected, and the image was subsequently rotated clockwise by the specified angle. The flipping operation involved the potential horizontal inversion of the image.

The final augmentation technique, cropping, entailed the random selection of a percentage within the range $[0, 0.5]$, which corresponded to the proportion of pixels to be removed from the sides of the image. It is important to note that each of these online augmentations was subject to a 50% probability of occurrence, and if none were selected through the random process, one transformation was manually chosen to ensure the application of at least one augmentation technique.

In light of the fact that the mentioned augmentation techniques are applied to a sequence of images, it is imperative to maintain consistency and accuracy throughout the entire sequence. To achieve this, the same augmentation operation is systematically performed on each image within the sequence, thereby ensuring uniformity and coherence across the series of images.

This uniform application of augmentations is crucial for preserving the spatiotemporal relationships and patterns present in the original sequence. By maintaining consistency across the augmented sequence, the analysis and subsequent modeling efforts can effectively leverage the inherent correlations and dependencies in the data. Moreover, this approach reduces the potential for introducing artifacts or distortions that could adversely impact the quality of the dataset and the reliability of the analysis.

5.3 Proposed method

In the context of this research, the method of stacking images as input for neural networks was chosen, primarily due to its ability to effectively capture and analyze the complex spatiotemporal relationships present in sequential spray images. The decision to use this approach was based on several key advantages that it offers, as detailed below.

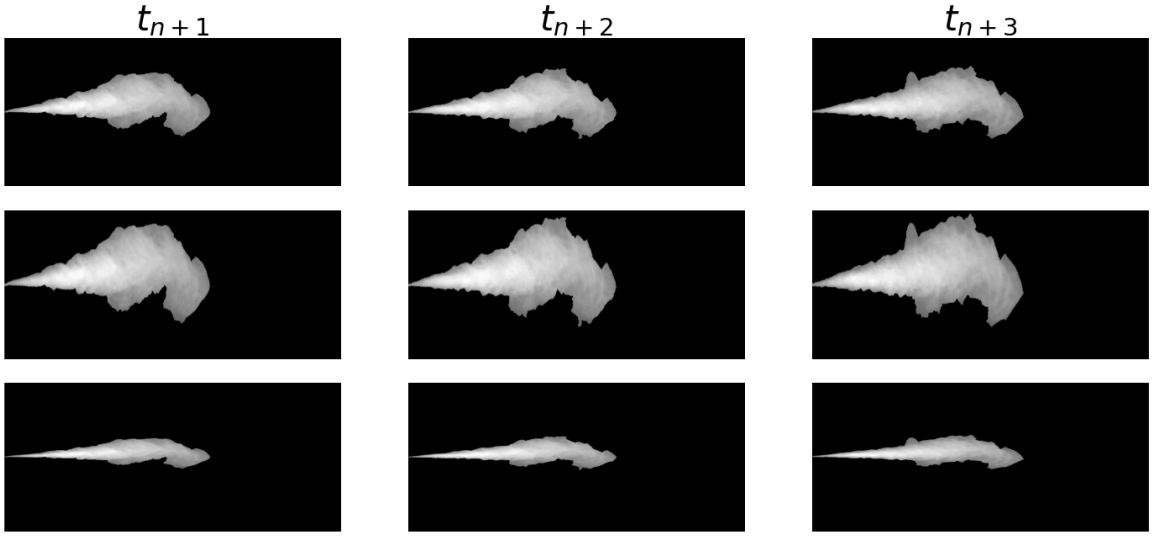

Figure 5.2: Illustration of a sequence of preprocessed images, showcasing variations in spray shapes. The first row exhibits the original sequence, while the second and third rows display the modified sequences with enlarged and reduced cone angles, respectively, achieved through stretching and compressing transformations.

Firstly, stacking images preserves the temporal context of the data, providing the neural network with simultaneous access to multiple instances across time, which is what we wanted to achieve with spray images gathered during one injection. This enables the model to effectively learn the inherent spatiotemporal patterns and relationships present in the sequence. Secondly, this method allows the network to extract spatial features from individual frames while considering the temporal context. Using one of the main strengths of convolutional neural networks (CNNs) in capturing local spatial patterns, the neural network can exploit both spatial and temporal features simultaneously, resulting in more accurate and robust performance. Moreover, stacking images as input simplifies the network architecture, reducing the need for recurrent or sequential processing layers that can introduce additional computational complexity. This leads to more efficient training and inference time. Furthermore, the use of stacked images as input enhances the robustness of the model by exposing it to a broader range of information.

Deep neural regression networks have emerged as a powerful tool for analyzing both single images and sequences of images. In this research, two distinct methods, namely StackNet and CNN-LSTM, were implemented to evaluate the performance of these networks in the context of image sequence analysis. Both methods comprise a feature extractor and a fully connected layer at the end, with the key distinction between them being the inclusion of a Long Short-Term Memory (LSTM) cell in the CNN-LSTM method. These approaches are designed to accept single images or sequences of images as input and generate a single output value representing the cone angle of the last image in the sequence.

Three state-of-the-art feature extractors were employed in this study: VGG, EfficientNet, and MobileNetV3, each offering unique advantages and capabilities.

The VGG16 configuration was selected as a representative of the VGG family. VGG builds upon the foundational concepts introduced in AlexNet [74], but opts for smaller 3x3 kernel sizes instead of the larger receptive fields employed in AlexNet. This choice results in fewer parameters due to the smaller receptive fields. Furthermore, VGG incorporates 1x1 convolutional layers, which enable the network to achieve non-linear decision functions without altering the receptive fields. The use of smaller convolution filters allows VGG to possess a greater number of weighted layers, ultimately contributing to enhanced performance. The architecture scheme can be seen on Fig. 5.3.

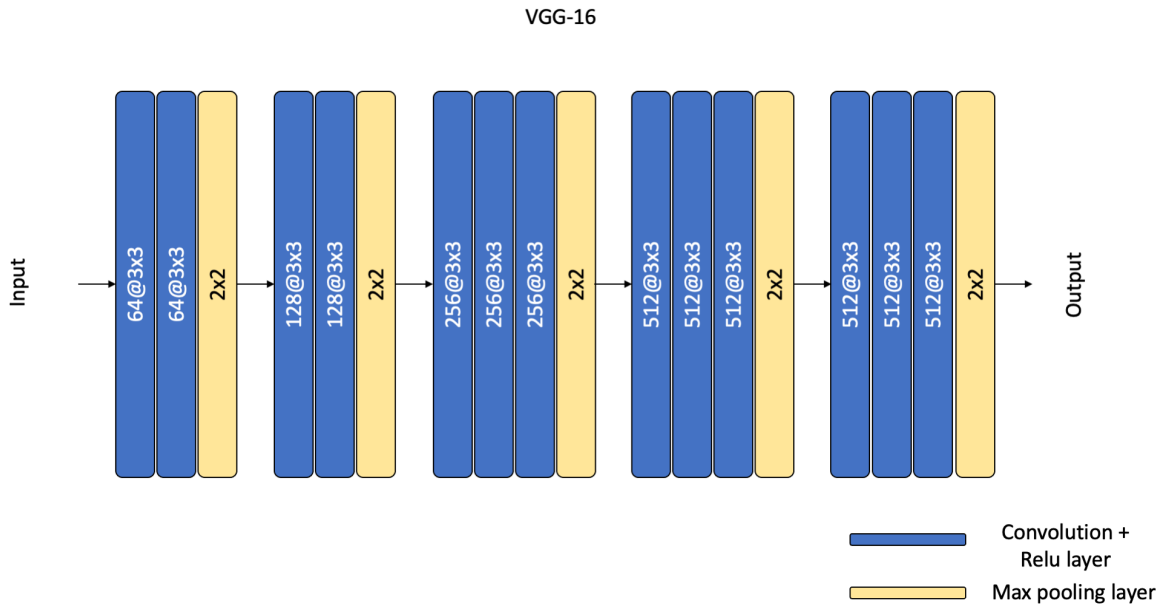

Figure 5.3: VGG16 architecture scheme

EfficientNet employs a technique known as compound coefficient scaling to scale models in a more effective manner [75]. Compound scaling uniformly scales each dimension using a fixed set of coefficients, rather than arbitrarily scaling width, depth, or resolution. This method is predicated on the notion of balancing the three dimensions of resolution, width, and depth by scaling them with a constant ratio. In this study, the EfficientNetB0 configuration was utilized.

MobileNetV3, the latest iteration of Google's MobileNet architecture [76], builds upon the innovations introduced in MobileNetV1 and MobileNetV2 [70, 77]. MobileNetV1 introduced depthwise separable convolutions, while MobileNetV2 further reduced network complexity and model size by employing an inverted residual structure, thus enabling more efficient deployment on mobile devices. Notably, non-linearity was removed from the narrow layers in the V2 model. MobileNetV3, in turn, is specifically tuned for mobile devices through the use of hardware-aware network architecture search (NAS) techniques. The architecture has been refined through various novel advancements, resulting in performance improvements on both

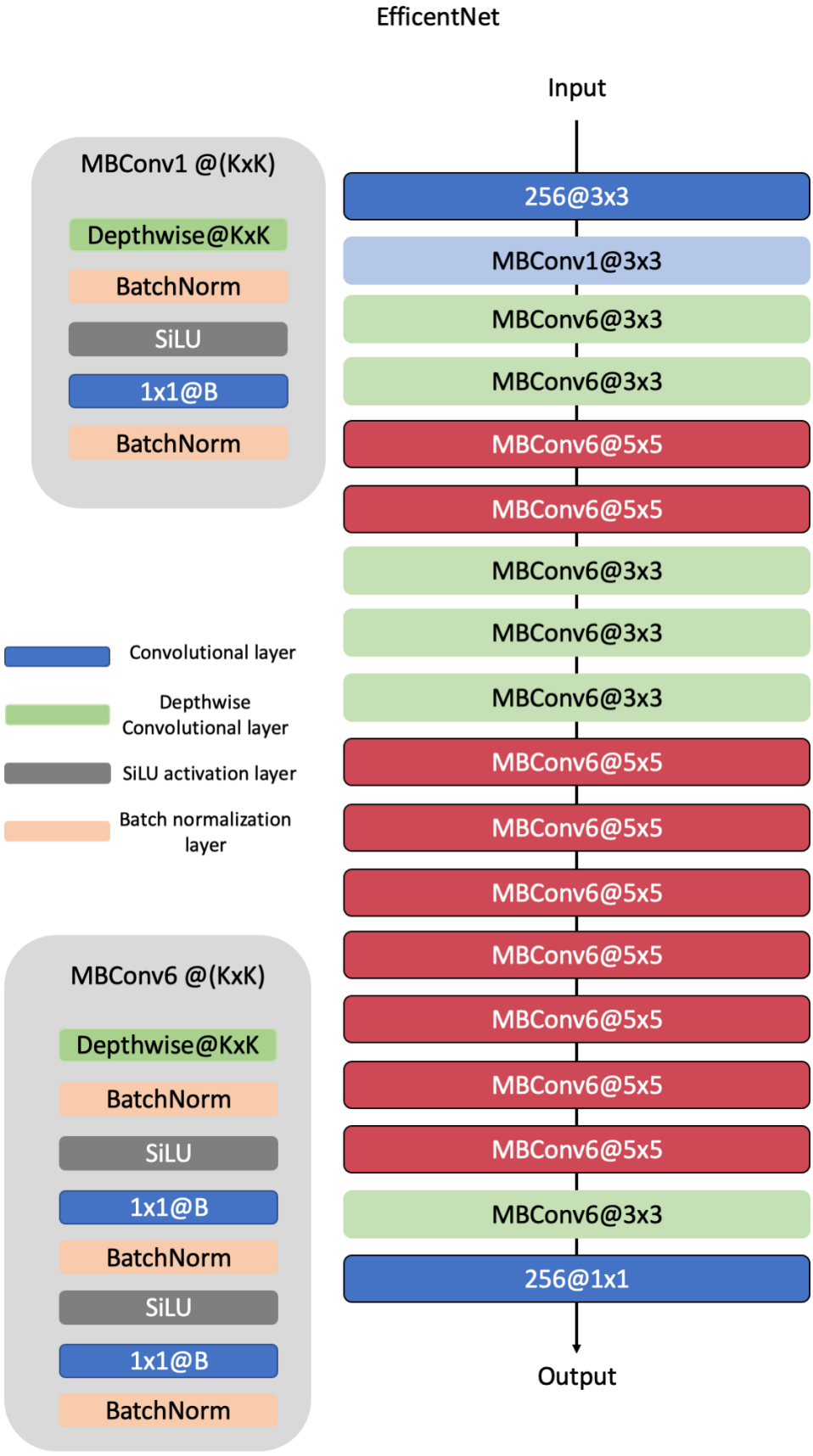

Figure 5.4: EfficientNet architecture scheme

ImageNet classification and Cityscapes segmentation tasks, while simultaneously reducing processing time. The architecture scheme is shown in Fig. 5.3.

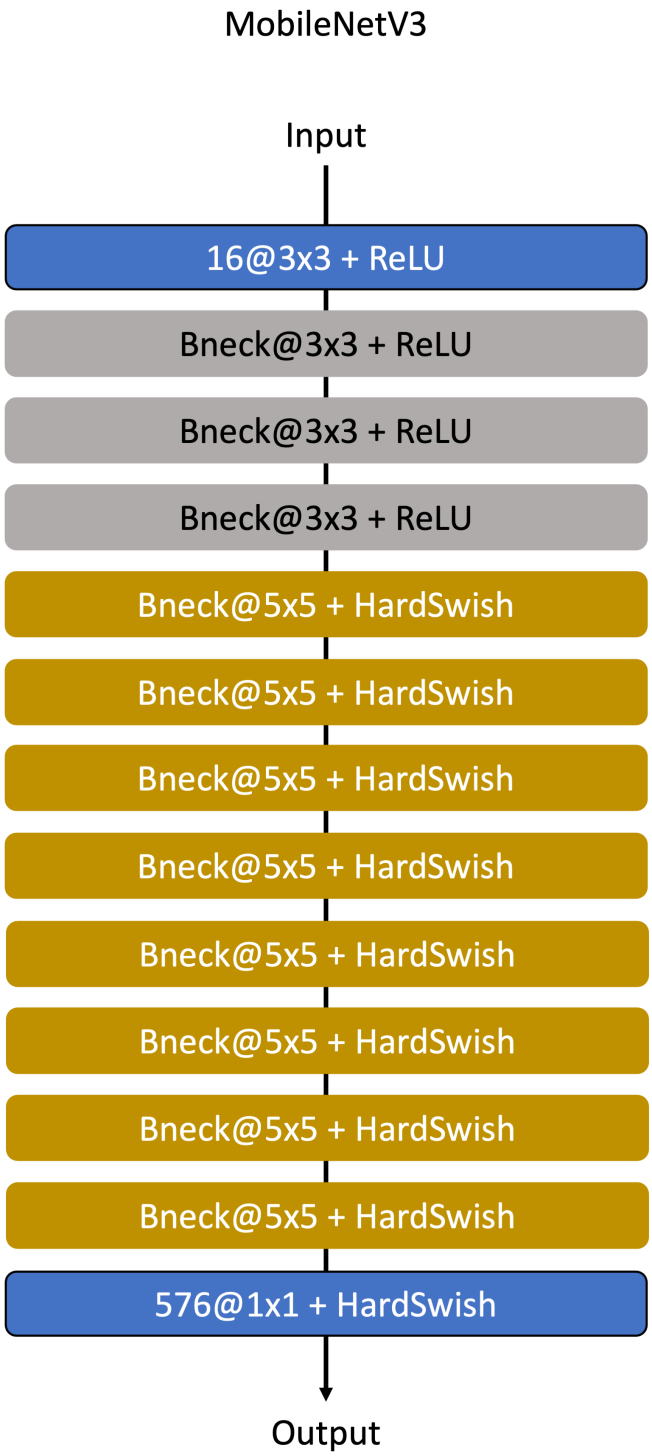

Figure 5.5: MobileNetV3 small architecture scheme: Bneck has two blocks with convolutional layers (1x1 and kernel size specified in block name), Batch Normalization, and activation function (ReLU or Hard Swish). Followed by a block with Adaptive Average Pooling and two convolutional layers with 1x1 kernel size and ReLU activation, and a final block with a 1x1 convolutional layer and Batch Normalization.

The fully connected layer is taken from the VGG original architecture. It consists of four layers. The first layer has 49× output channels from the final convolution from the feature extractor number of neurons, the second and third have 4096 neurons and the final one has 1 because we have a regression model. The number of neurons in the first layer for each feature extractor is shown in Table 5.2.

Table 5.2: Number of neurons in the first fully connected layer for each of the three mentioned state-of-the-art feature extractors

Backbone	Number of neurons in first FC layer
VGG	25 088
EfficientNet	28 224
MobileNetV3	62 720

5.3.1 StackNet

StackNet is a neural network architecture that made of a feature extractor and a fully connected regressor. The network accepts a sequence of consecutive grayscale spray images as input. In contrast to traditional approaches, which involve using a batch of single RGB images with input dimensions of $[batch\ size, channels, width, height]$, StackNet inputs dimensions of $[batch\ size, t, width, height]$, where t defines the number of sequence images uses. As discussed in Section 5.3, the feature extractor can be one of three architectures: VGG16, EfficientNetB0, or MobileNetV3. Following the feature extraction process, the feature map is passed through an Adaptive Average Pooling layer with an output size of 7×7, which is utilized to reduce the dimensionality of the feature map.

The regressor part of the network consists of a fully connected layer with three sublayers. The input to the first layer consists of a flattened tensor containing a number of features equal to $7 \times 7 \times$ the number of final kernels from the feature extractor. For VGG16, this number is 512, while it is 576 for MobileNetV3 and 1280 for EfficientNetB0. These numbers are displayed in Table 5.2. The first and second layers consist of 4096 features, and the final layer comprises a single feature for regression. A ReLU activation function is applied after each layer, and Dropout with a probability of 0.5 is employed for regularization purposes. The architecture of StackNet is depicted in Fig. 5.6.

5.3.2 CNN-LSTM

Long Short-Term Memory (LSTM) networks were introduced in 1997 [78], and they are a form of recurrent neural networks (RNN) architecture. The idea behind LSTMs is the cell state,

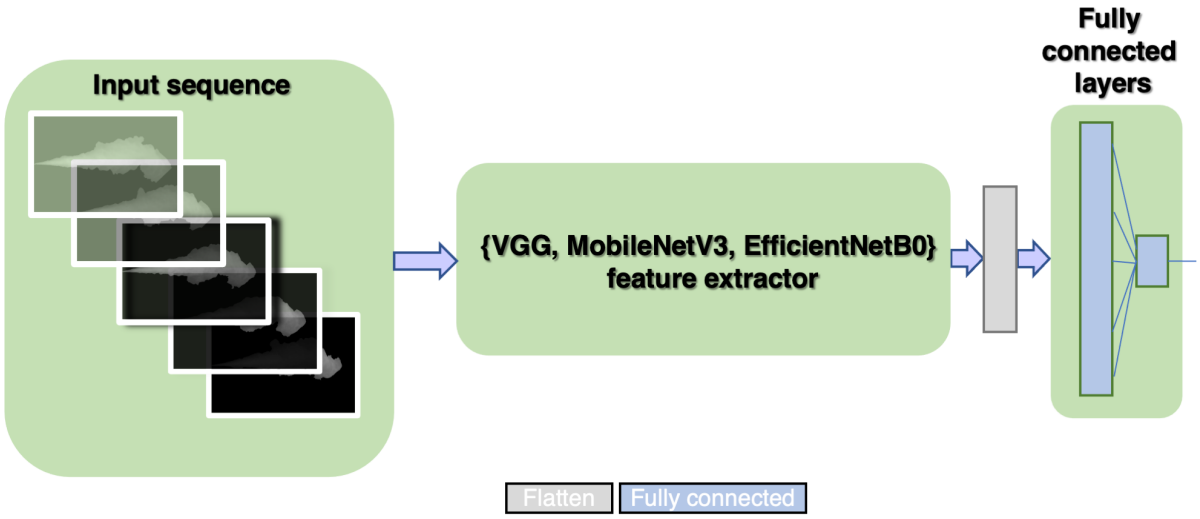

Figure 5.6: StackNet architecture

represented by a horizontal line traversing the top of the LSTM unit, which facilitates the flow of information across multiple time steps. This design allows LSTMs to effectively learn and retain long-term dependencies within the input data. The cell state is regulated by three distinct gates: the input, forget, and output gates, which collectively determine the addition, removal, or output of information from the cell state at each time step. The input gate is responsible for identifying which values from the input should be incorporated into the cell state, while the forget gate dictates which values in the cell state should be eliminated or "forgotten." Lastly, the output gate declares the information from the cell state to be output at the current time step. Each of the three gates are controlled by sigmoid activation functions, which produce values ranging between 0 and 1, indicating the degree to which the gates are open or closed. The LSTM cell is illustrated in Fig. 5.7.

The CNN-LSTM concept was originally proposed in [79] for generating captions from a sequence of images, as LSTMs are mostly utilized in natural language processing (NLP) applications. In the thesis, the CNN-LSTM architecture will be implemented for the purpose of spray cone angle estimation. The CNN-LSTM model utilizes a feature extractor as the backbone to extract feature maps from input images. Once the feature map is obtained, it is forwarded to the LSTM layer, and fed into the fully connected layer. CNN-LSTM shares similarities with StackNet; however, it integrates an LSTM layer after flattening the feature map and prior to the fully connected layer. The LSTM layer is stacked, comprising two LSTM cells, each with 128 features in the hidden state. The feature extractor can be one of three architectures: VGG16, EfficientNetB0, or MobileNetV3, the same as in StackNet. The regressor is also fully connected, consisting of three layers with ReLU activation and Dropout. The primary distinction is that the first layer contains 128 features, which corresponds to the output of the last LSTM cell, as opposed to StackNet, which possesses 49 $\times$ the number of kernels from the final layer of the

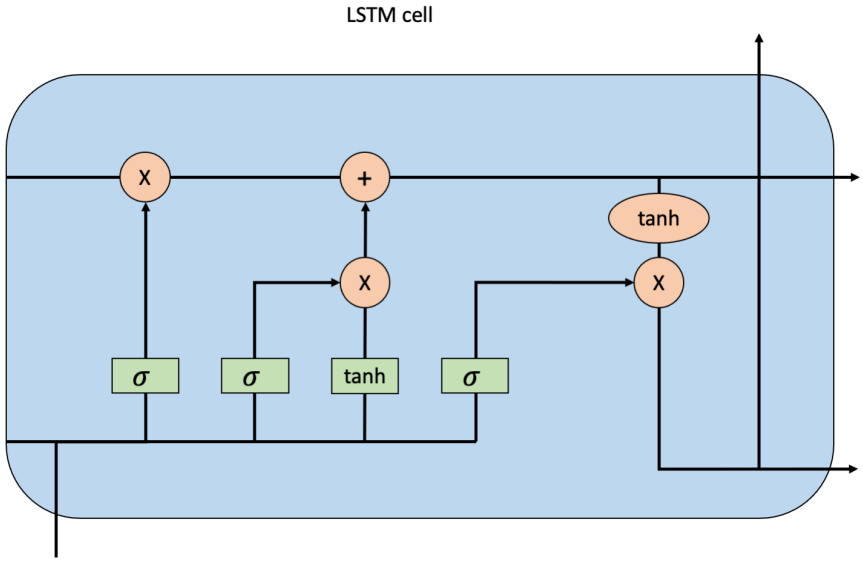

Figure 5.7: LSTM cell diagram. The leftmost sigmoid function represents the forget gate F_t, the middle sigmoid and lower tanh functions together represent the input gate I_t, and the rightmost sigmoid function denotes the output gate O_t. The symbol 'X' signifies multiplication, while the '+' symbol indicates concatenation.

feature extractor. The architecture of CNN-LSTM is shown in Fig. 5.8.

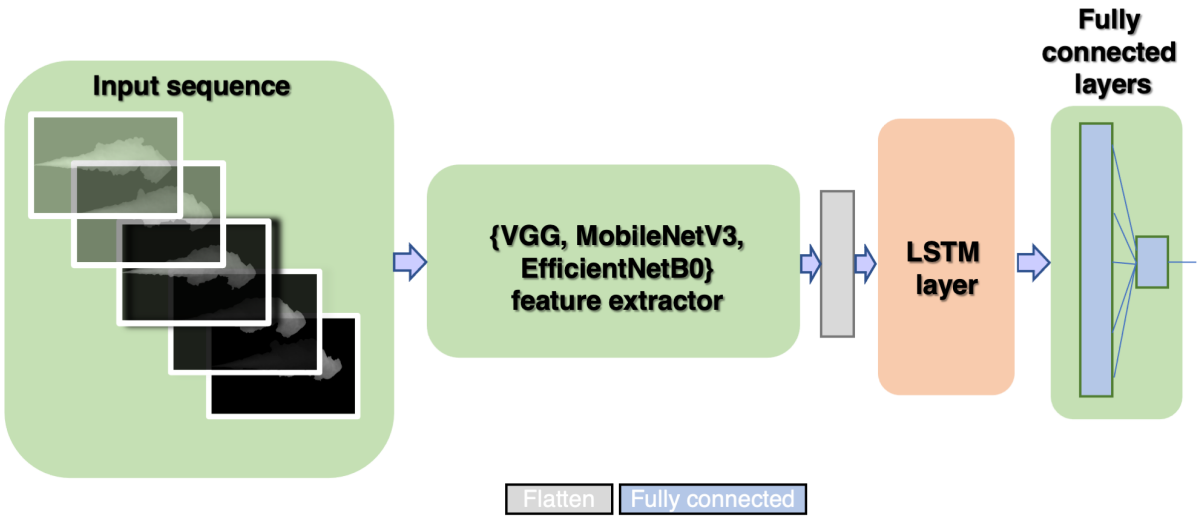

Figure 5.8: CNN-LSTM architecture

5.3.3 Extended StackNet

The original StackNet architecture explained in Sec. 5.3.1 consists of a feature extractor which is responsible for capturing the essential characteristics of the images, while the fully connected

layer aims to map the extracted features to the cone angle estimation. However, the idea behind the Extended StackNet is that the original architecture lacks the ability to fully understand the temporal relationship between the images in the sequence, which could be crucial for accurate cone angle estimation.

By adding a parallel fully connected layer, the network can simultaneously process information about the previous cone angles from the sequence. This layer effectively acts as a memory component, enabling the model to learn and recognize patterns in the cone angles across the whole sequence. Because of it, it becomes more sensitive to changes in the cone angles and can make better predictions. Moreover, incorporating the previous angles context improves the network's capacity. By having a better understanding of the relationships between the cone angles in the sequence, the network can better distinguish between actual trends and random fluctuations, leading to more robust and accurate predictions. The output of the parallel FC layer is concatenated to the flattened feature map and fed to the original FC layer.

Four different variations of Extended StackNet were tested:

- Basic StackNet with added fully connected layer (EStackNet): This architecture is an extension of the StackNet architecture described in Sec. 5.3.1, incorporating an additional parallel three fully connected layers. The first FC layer has an input of $t-1$, where t represents the number of sequence images and an output of 128. The second layer consists of 256 neurons, while the last layer has 1024 neurons. To reduce the model complexity, a final convolution with 128 kernels of size 3×3 is added to the feature extractor. This way, fewer neurons are needed in the fully connected part which is the most computationally demanding part of the network.
- EStackNet3D: This architecture employs 3D convolutions within the feature extractor, which are commonly used in medical imaging and video analysis. Since the problem involves a sequence of images, 3D convolutions were taken into account as a potential solution. Although 3D convolutions have demonstrated better feature extraction capabilities compared to 2D convolutions by capturing relationships across all three dimensions, they come at a higher computational cost and increased memory usage. EStackNet3D's feature extractor utilizes two 3D convolutions, each followed by 3D max pooling and a ReLu activation function. The first convolution has 32 kernels, while the second has 64 kernels, each with a kernel size of $3 \times 3 \times 3$. Both max pooling layers have a kernel size of $1 \times 2 \times 2$ with a stride of $1 \times 2 \times 2$. The parallel fully connected layer contains three layers, with the first having 128 neurons, the second 256, and the final one 1024.
- MiniEStackNet - This architecture's feature extractor consists of four identical blocks, varying only in the number of kernels in the convolution. Each block has two convolutional layers, followed by ReLu and a max pooling layer. All convolutions have a kernel size of 3×3, while all max pooling layers have a kernel size of 3×3 with a stride of 3.

The feature extractor follows the encoder pattern of the U-Net architecture. The first convolution has 32 kernels, the next two have 64, the fourth and fifth have 128, and the last three have 256 kernels. With this configuration of max pooling layers, an input image of size 256×256 pixels will result in a feature map with dimensions of $1 \times 1 \times 256$, where 256 corresponds to the number of kernels in the last convolutional layer. The parallel FC layer has only two layers, with the first having 128 neurons and the output having 256 neurons, the same as the dimension of the feature map. The input shape to the regressor of the entire architecture is $256 + 256$, and it has three layers with 256, 128, and 1 neurons. It is referred to as "Mini" due to its more lightweight design and fewer parameters compared to the previously proposed methods.

•MiniEStackNet16 - This architecture is similar to MiniEStackNet, but with a reduced number of kernels in the feature extractor and fewer neurons in the parallel FC layer. It is even more lightweight, with the initial number of kernels being 16, hence the name. The second and third convolutions have 32 kernels, while the last five convolutions have 64 kernels. To further minimize the model, both layers in the parallel FC layer have 32 neurons. The regressor has the same architecture as MiniEStackNet.

5.4 Experimental results

A series of experiments were conducted to identify the most accurate model for the task at hand. The deep neural networks utilized in these experiments were trained on augmented images, as detailed in Section 5.2.2. To evaluate the performance of the models, the Mean Absolute Error (MAE) was employed as the evaluation metric, as illustrated in Eq. 5.2.

$$MAE = \frac{1}{N} \sum_{i=1}^{N} |\hat{y}_i - y_i| \tag{5.2}$$

In the equation above, $\hat{y}_i$ represents the estimated cone angle, y_i stands for the ground truth value for the cone angle, and N marks the total number of elements considered in the evaluation process. By using MAE as the evaluation metric, we aimed to quantify the average discrepancy between the predicted and actual cone angles, which serves as a reliable indicator of the model's performance.

All the methods under investigation were trained, validated, and tested using the previously mentioned dataset. The Adam optimizer, coupled with a cyclic learning rate scheduler with one cycle, was employed during the training process [80]. For both StackNet and CNN-LSTM models, a base learning rate of 10^{-5} and a maximum learning rate of 10^{-3} were set. On the other hand, the Extended StackNet models were trained with a base learning rate of 10^{-7} and a maximum learning rate of 10^{-4}. The loss function utilized throughout the training process was

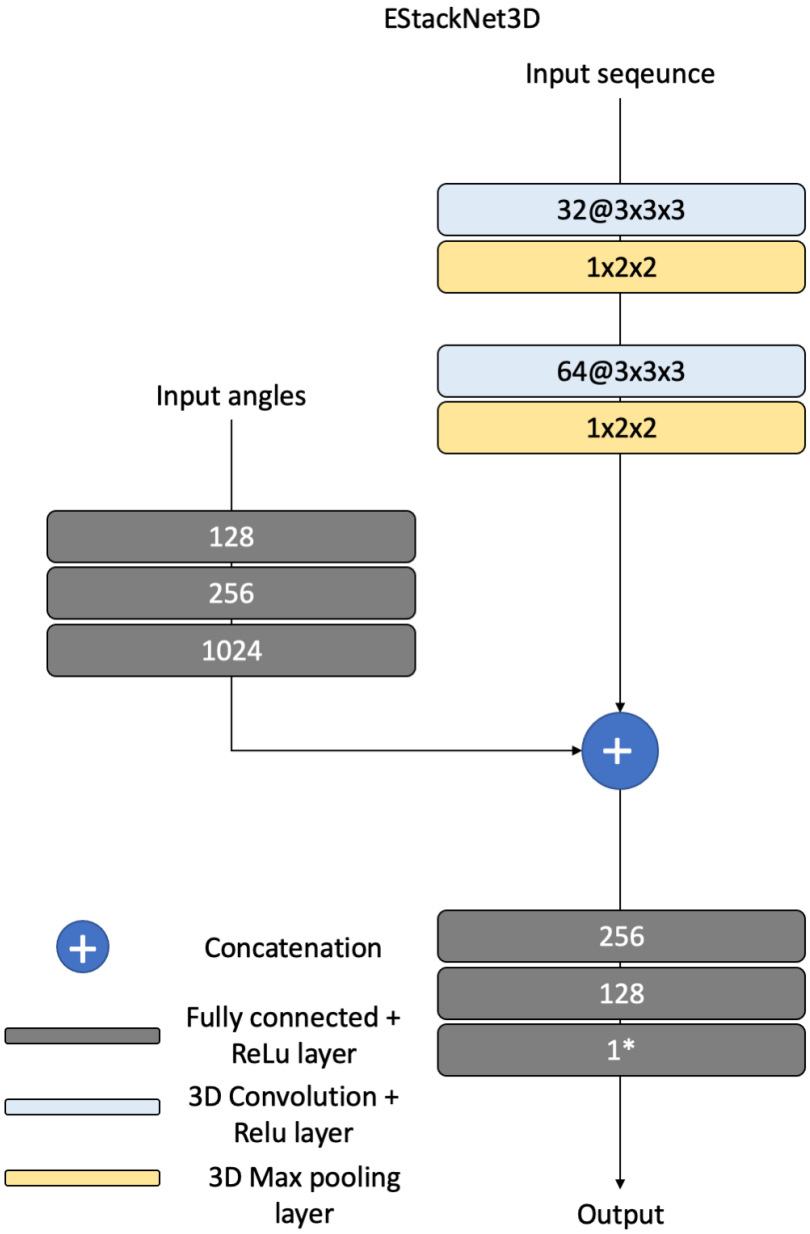

Figure 5.9: EStackNet3D architecture

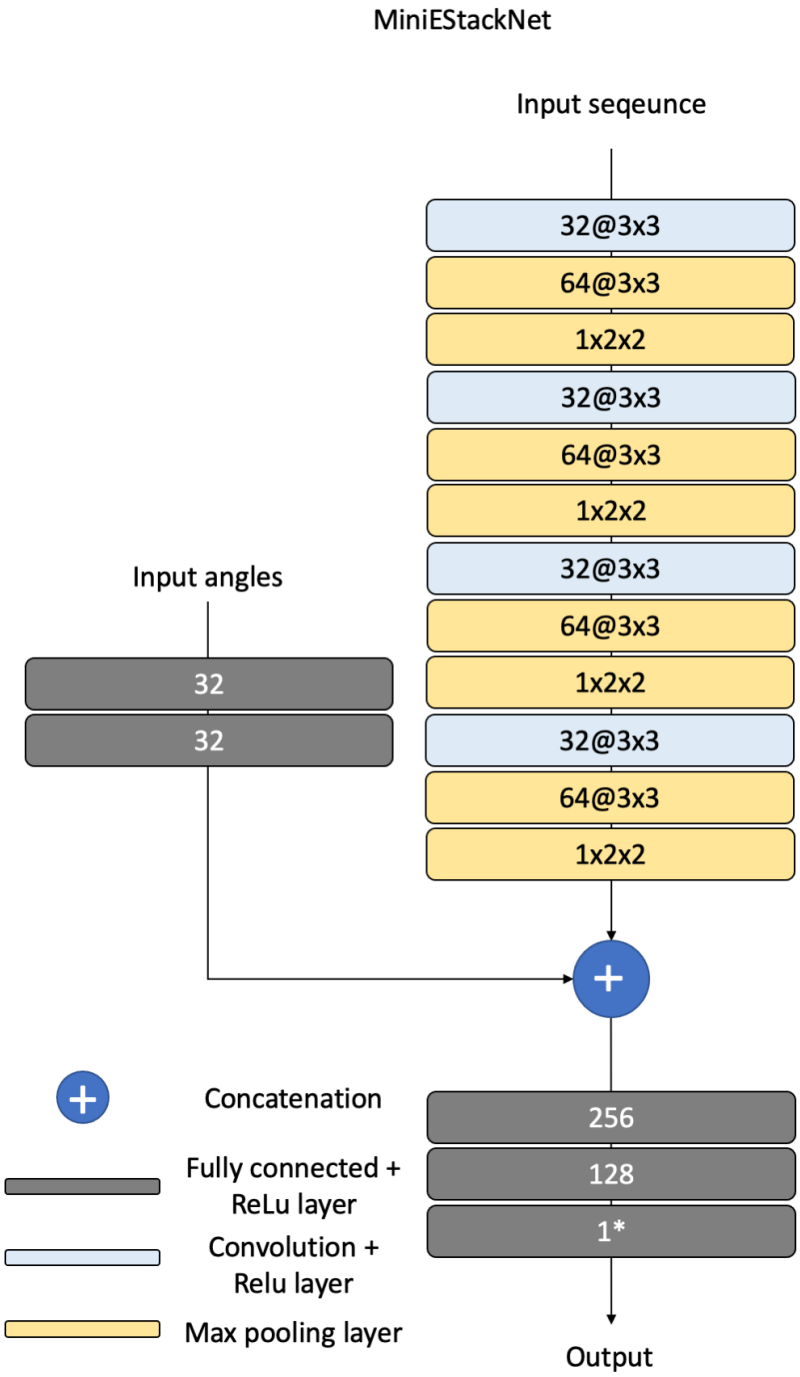

Figure 5.10: MiniEStackNet architecture

the Mean Squared Error (MSE) between the predicted outputs and the corresponding ground truth labels, as depicted in Eq. (5.3).

$$Loss = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2 \tag{5.3}$$

In the equation above, $\hat{y}_i$ represents the model's output, y_i denotes the ground truth value, and N denotes the total number of elements considered. The MSE loss function measures the average squared difference between the predicted values and the actual values, thereby providing an effective means of quantifying the model's performance during the training phase.

The Mean Absolute Errors (MAE) for both StackNet and CNN-LSTM models are presented in Table 5.3. These results will be compared to a baseline approach, in which the three before mentioned feature extractors take only a single image as input, thereby not accounting for the sequence of images. The baseline results can be found in the first row of Table 5.3, denoted by t=1, where 't' represents the number of images in the input sequence being utilized for the estimation process.

Table 5.3: A comparison of Mean Absolute Error (MAE) results between baseline models and the proposed StackNet and CNN-LSTM models. The first row indicates the number of images utilized in the input sequence, with the baseline approach corresponding to models that employ only a single image as input.

	t = 1	t = 2		t = 3		t = 4		t = 5		Backbone average
	Baseline	StackNet	CNN-LSTM	StackNet	CNN-LSTM	StackNet	CNN-LSTM	StackNet	CNN-LSTM	
EfficientNet	1.173	2.156	2.895	1.190	0.986	2.219	1.116	2.840	2.830	1.934
MobileNet	1.881	2.381	2.214	1.980	1.054	1.964	1.064	1.597	1.537	1.741
VGG	2.515	0.695	0.925	**0.505**	0.868	0.621	0.868	0.566	1.275	**0.982**
Model average	1.856	1.744	2.011	1.225	**0.969**	1.601	1.016	1.668	1.881	
t average	1.856	1.878		**1.097**		1.309		1.774		

5.4.1 StackNet results

Upon examining the input of two images (t=2), the StackNet using EfficientNet as a feature extractor achieves a mean absolute error of 2.156 degrees, while the MobileNet and VGG-based StackNets achieve errors of 2.381 and 0.695 degrees, respectively. For t=3, the StackNet with EfficientNet exhibits a mean absolute error of 1.190 degrees, the MobileNet-based StackNet 1.98 degrees, and the VGG-based StackNet 0.505 degrees. With four images as input (t=4), the mean absolute errors for StackNets with EfficientNet, MobileNet, and VGG are 2.219,

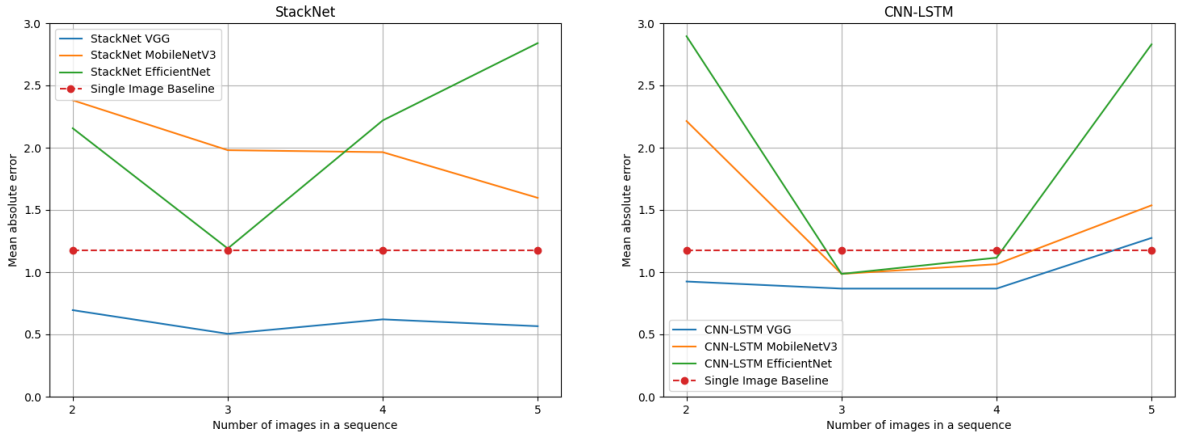

Figure 5.11: Comparative performance of StackNet (left graph) and CNN-LSTM (right graph) utilizing different backbones, contrasted with the single-image EfficientNet, which is denoted as the best baseline model.

1.964, and 0.621 degrees, respectively. Lastly, for a sequence of five images as input, the mean absolute errors for StackNets with EfficientNet, MobileNet, and VGG are 2.840, 1.597, and 0.566 degrees, respectively.

The StackNet with VGG as a feature extractor surpasses the performance of all baseline approaches and achieves the lowest error (0.505 degrees) among all tested models when t=3, as it can be seen in the left graph of Fig. 5.11. StackNet with EfficientNet demonstrates accuracy, outperforming the single-image VGG baseline model for every t value and exceeding the single-image MobileNet only when t=2. The MobileNet-based StackNet outperforms the single-image VGG for every t and surpasses the baseline MobileNet for t=5, but exhibits a higher error in all other cases. StackNet achieves the lowest average backbone error when the sequence length is three, with an error of 1.225 degrees. The average mean absolute error for all StackNet models is 1.6 degrees.

5.4.2 CNN-LSTM results

CNN-LSTM generally exhibits superior performance compared to StackNet, as evidenced by its lower mean absolute error (MAE) of 1.463 across all models, representing a reduction of 0.164 in error. However, CNN-LSTM does not achieve the smallest global error. Similar to StackNet, when utilizing two sequence images (t=2), CNN-LSTM with VGG yields a smaller error than all baseline models. With a sequence length of t=3, CNN-LSTM achieves the best average results across backbones for all models, with an MAE of 0.969, as shown in Table 5.3. For t=3, all backbones in CNN-LSTM outperform the best baseline model, effectively surpassing all baseline models. The same outcome occurs when t=4, though with a slightly larger average error (by 0.047). At t=5, CNN-LSTM with VGG achieves the best result of

1.275, although it outperforms only two of the three baseline approaches, falling short of the best baseline model with an MAE of 1.173. The performance of CNN-LSTM with different backbones can be observed in Fig. 5.11, displayed in the right graph.

5.4.3 Feature extractor results

The performance of all three backbone architectures has been thoroughly evaluated in this study. On average, VGG achieves the lowest mean error across all models, with a value of 0.982. This is followed by MobileNet, which obtains an error of 1.741, and finally, EfficientNet with the highest error of 1.934.

A comparison between the baseline approach using VGG and the proposed StackNet and CNN-LSTM models with VGG as the backbone reveals that both StackNet and CNN-LSTM significantly outperform the baseline model across all sequence image counts. This observation is evident in the left graph presented in Fig. 5.12. The middle graph in Fig. 5.12 displays the performance comparison of models using MobileNet as the backbone. For a sequence length of 2, the baseline model with MobileNet demonstrates better accuracy than both StackNet and CNN-LSTM. However, for sequence lengths of 3 and 4, CNN-LSTM achieves a larger error than the baseline approach, while StackNet surpasses both in terms of accuracy. In the final case, with a sequence length of 5, both proposed methods estimate the cone angle with greater accuracy than the baseline approach.

EfficientNet was also assessed in this study. The baseline approach with EfficientNet, using a single image, exhibits the best accuracy among all baseline models. When the sequence length is 2 and 5, it outperforms both StackNet and CNN-LSTM with EfficientNet as the backbone. Nonetheless, for sequence lengths of 3 and 4, it estimates the cone angle more accurately than StackNet alone. These findings are illustrated in the right graph in Fig. 5.12.

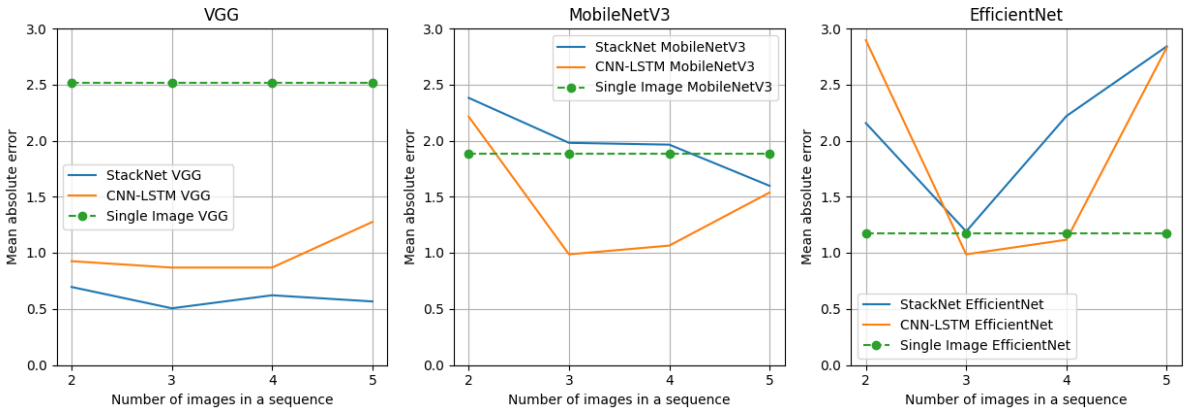

Figure 5.12: Performance comparison of the evaluated backbones against single-image models, with the first graph representing VGG backbones, the second graph showcasing MobileNetV3, and the third graph featuring EfficientNet.

5.4.4 Extended StackNet results

The Extended StackNet models demonstrated superior performance compared to the standard StackNet and CNN-LSTM models. The results can be found in Table 5.4, which presents the mean absolute errors of the Extended StackNet model variants alongside the averages for each model across the sequence lengths, as well as the averages for sequence lengths across the models. Table 5.5 displays the number of parameters for each variant and their corresponding averages for every sequence length, as the parameter count varies depending on the number of input images. Fig. 5.13 shows the comparison of all EStackNet variations when taking into account the average mean absolute error and number of parameters.

Table 5.4: Comparison of Mean Absolute Error (MAE) for Extended StackNet method variants

	t = 2	t = 3	t = 4	t = 5	Model average
EStackNet - VGG	0.562	0.512	0.517	0.521	**0.528**
EStackNet - EfficientNet	0.854	0.513	0.727	0.627	0.680
EStackNet - MobileNet	0.711	0.882	0.817	0.841	0.813
EStackNet3D	0.752	0.562	0.611	0.577	0.626
MiniEStackNet	0.572	0.497	**0.476**	0.662	0.552
MiniEStackNet16	0.697	0.844	0.509	0.945	0.749
t average	0.691	0.641	**0.610**	0.696	

The VGG-based Extended StackNet, with added fully connected layers, has proven to be the most effective among all the basic StackNet models. Its best result, with a mean absolute error of 0.512 degrees, is achieved for a sequence length of 3, and it is already comparable to the best global results of basic StackNet and CNN-LSTM (0.505 degrees). For sequence lengths of 2, 4, and 5, the VGG-based Extended StackNet yields mean absolute errors of 0.562, 0.517, and 0.521 degrees, respectively. The smallest average error, 0.528 degrees, can be attributed to its substantial number of parameters, averaging 70.139 million, which is the largest among the classic Extended StackNet variations.

The EfficientNet-based Extended StackNet outperforms the baseline EfficientNet methods as well as the classic StackNet and CNN-LSTM models with EfficientNet across all sequence lengths. It produces mean absolute errors of 0.854, 0.513, 0.727, and 0.627 degrees for sequence lengths of 2, 3, 4, and 5, respectively. This model has around 10 million fewer parameters than the VGG-based Extended StackNet.

Regarding the Extended StackNet with MobileNet, which possesses the fewest parameters

of all classic Extended StackNet models, a similar pattern emerges as with the classic StackNet and CNN-LSTM models with MobileNet. The average error is the largest of all tested Extended StackNet methods at 0.813 degrees. The model produces errors of 0.711, 0.882, 0.817, and 0.841 degrees for sequence lengths of 2, 3, 4, and 5, respectively.

The EStackNet3D model, on average, yields better results than the EfficientNet-based and MobileNet-based Extended StackNet models but underperforms the VGG-based Extended StackNet by 0.098 degrees. It achieves its smallest error, 0.562 degrees, for a sequence length of 3, and records errors of 0.752, 0.611, and 0.577 degrees for sequence lengths of 2, 4, and 5, respectively. EStackNet3D is also the most complex of all Extended StackNet models, with an average parameter count exceeding 210 million. This complexity can be attributed to its 3D convolutions.

The MiniEStackNet model achieves the lowest global error of all tested models at 0.476 degrees and surpasses the best classic StackNet with VGG as shown in Table 5.3. It records the second-best average error, trailing by only 0.024 degrees (4.54%), while having significantly fewer parameters (28 times less), rendering it the most effective model overall considering its complexity and performance. The mean absolute errors obtained by MiniEStackNet for sequence lengths of 2, 3, 4, and 5 are 0.572, 0.497, 0.476, and 0.662 degrees, respectively. In contrast, the MiniEStackNet16 model achieves the second-worst results with an average error of 0.749 degrees. However, it is also the most lightweight model, featuring only 380 thousand parameters.

Similar to the classic StackNet and CNN-LSTM methods, the Extended StackNet models achieve the largest average mean absolute errors when processing sequences with lengths of 2 or 5 images, outputting errors of 0.691 and 0.696, respectively. However, it is crucial to note that the errors associated with these sequence lengths are still smaller by a factor of approximately 3 when compared to the corresponding errors in the classic StackNet and CNN-LSTM methods.

In the case of the classic StackNet and CNN-LSTM approaches, the smallest average error is observed for a sequence length of 3 (1.097 degrees). This trend does not hold true for the Extended StackNet models, which demonstrate the best performance with a sequence length of 4, achieving a mean absolute error of 0.610 degrees. The second-best error is obtained when the sequence length is 3, with a value of 0.641 degrees.

The Extended StackNet methods exhibit a more consistent performance across different sequence lengths compared to the classic StackNet and CNN-LSTM approaches. This observation can be seen from the relatively small difference between the best and worst sequence length results, which amounts to 0.086 degrees in the case of the Extended StackNet models. In contrast, the difference for the classic methods is significantly larger, at 0.781 degrees.

In summary, the Extended StackNet models exhibit improved performance compared to the standard StackNet and CNN-LSTM models. Among the Extended StackNet models, the

Table 5.5: Comparison of the number of parameters for Extended StackNet method variants expressed in millions

	t = 2	t = 3	t = 4	t = 5	Model average
EStackNet - VGG	70.139	70.139	70.140	70.141	70.139
EStackNet - EfficientNet	60.317	60.317	60.317	60.318	60.137
EStackNet - MobileNet	56.425	56.425	56.426	56.426	56.426
EStackNet3D	134.866	201.975	269.084	336.193	210.529
MiniEStackNet	2.475	2.475	2.476	2.476	2.476
MiniEStackNet16	0.380	0.380	0.380	0.380	0.380

VGG-based variant demonstrates the best performance in terms of error rate, which can be attributed to its large number of parameters. EfficientNet-based and MobileNet-based Extended StackNet models also show enhanced results compared to their respective baseline methods. The EStackNet3D model exhibits increased complexity due to its 3D convolutions, while the MiniEStackNet model offers an optimal balance between performance and complexity, achieving the lowest global error and the second-best average error, which can clearly be seen in Fig. 5.13. Lastly, although the MiniEStackNet16 model yields the second-worst results, it serves as the most lightweight option with a significantly lower parameter count.

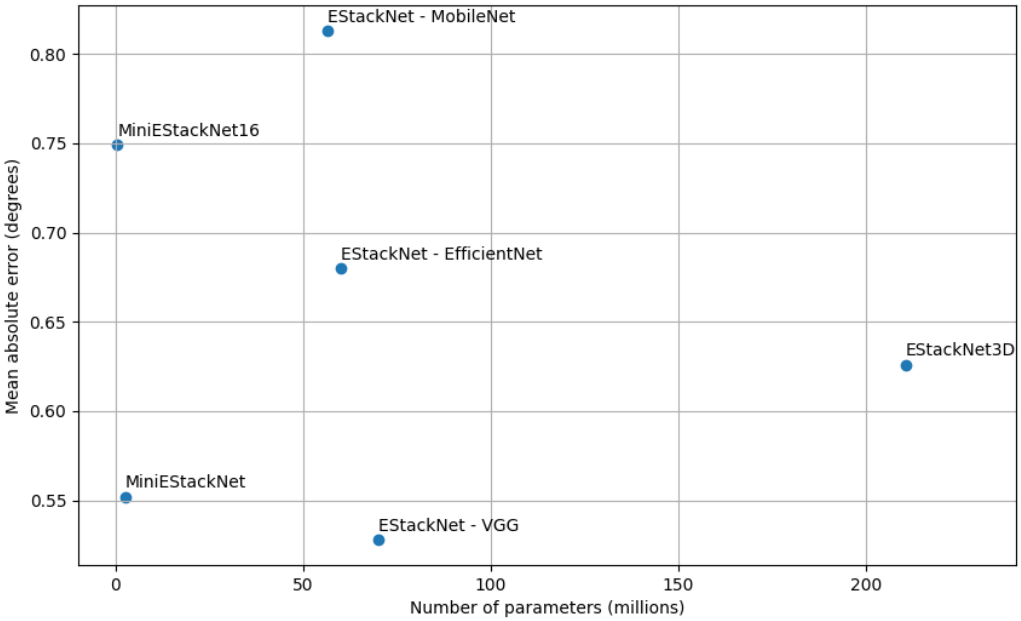

Figure 5.13: Comparision of EStackNet variants when looking at average mean absolute error and average number of parameters

Chapter 6

Conclusion

Macroscopic spray parameters, including cone angle, spray penetration length, and spray area, play a critical role in the efficiency and performance of internal combustion engines. Optimizing these parameters is essential for developing more environmentally friendly and cleaner engines, particularly in heavy-duty transportation applications where internal combustion engines are most commonly used. Accurate macroscopic spray parameters can be utilized as input data for numerical simulations aimed at estimating engine efficiency and reducing pollutant emissions.

To acquire these parameters, high-speed cameras are employed, capturing spray images that are then subjected to computer vision algorithms. A majority of the existing methods for determining spray macroscopic parameters rely on traditional, non-learning-based approaches. Although these methods have demonstrated efficacy, they suffer from certain limitations, such as a lack of generalization across various spray image types and the necessity for manual adjustment of hyperparameters.

Given the increasing popularity and success of deep learning-based techniques, the field of computer vision has experienced significant advancements, making it well-suited for addressing the challenges of spray image analysis by providing more robust, general, and accurate solutions. This thesis focuses on determining spray macroscopic parameters through segmentation-based and regression-based approaches, both employing deep learning-based methodologies. Two primary methods are proposed.

The first method utilizes a lightweight deep neural network called Min U-Net, which is based on the state-of-the-art segmentation model U-Net. An ablation study was conducted to identify the optimal Min U-Net architecture, exploring the depth of the network and the number of kernels in the convolutional layers. The model was trained on a small dataset of 200 images, which was expanded through data augmentation, effectively increasing the training set from 120 images to 12,000 images. Min U-Net outperforms traditional non-learning-based methods such as thresholding, the Otsu method, and max entropy segmentation. Furthermore, it achieves competitive results in comparison to other state-of-the-art deep learning models, while

possessing between 500 and 5,600 fewer parameters. With only 4.4 thousand parameters, Min U-Net achieves a mean Dice score of 0.95 and a median of 0.96. The model also processes images with a faster inference time of 11.94 ms/image, more than twice as fast as the quickest state-of-the-art baseline models. The proposed method exhibits higher accuracy in estimating spray macroscopic parameters compared to other approaches found in the literature, achieving a mean average error of 1.08 degrees for cone angle estimation and mean relative errors of 5.95% and 4.05% for spray penetration and spray area, respectively.

The second method involves a regression-based deep neural network that employs a sequence of images as input instead of a single image. This approach is motivated by the fact that images are captured during an injection, suggesting that a sequence of images could produce better results than a single image. The original cone angle values show a small standard deviation of only 1.74 degrees. To address overfitting and enhance generalization, data augmentation techniques, both online and offline, were employed. Offline augmentation is performed prior to training, while online augmentation occurs during the batching of inputs from the dataset. Spray images were preprocessed using rotation and shifting techniques to produce equally situated images, which were then widened and narrowed to generate a broader range of cone angles. This augmentation increased the label variance from 1.74 degrees to 12.91 degrees and expanded the cone angle range from $[17.25, 26.03]$ to $[5.09, 58.52]$, while the mean and median remained constant. Additionally, the data augmentation increased the dataset size from 196 to 3,276 images. During training, images were also subjected to random rotations, flips, and crops.

Two proposed methods, StackNetand CNN-LSTM, utilize a feature extractor in conjunction with a fully connected layer. The CNN-LSTM method also incorporates an LSTM layer between the feature extractor and the fully connected layer, while StackNet does not. StackNet and CNN-LSTM were evaluated using three different feature extractors: VGG16, MobileNetV3, and EfficientNetB0. Both methods employ a sequence of images as input and feature extraction with a fully connected layer at the end, with the distinguishing factor being the presence of an LSTM layer in the CNN-LSTM method. These methods were compared to baseline approaches that use a single image as input.

The lowest global mean absolute error was achieved using the StackNet and VGG combination, with an error of 0.505 degrees. The smallest average model error for CNN-LSTM was 0.969 degrees when the sequence length of images was three. Moreover, this method outperformed all baseline approaches by over 17%. The lowest average error observed for sequence length alone was 1.097 degrees, with a sequence length of three images. VGG emerged as the most accurate backbone, producing an average error of 0.982 degrees when compared to all backbones. These results indicate that the optimal number of sequence images for accurate estimation of the cone angle is three and that models generally perform better when utilizing an image sequence as opposed to a single image.

To further enhance the StackNet and CNN-LSTM methods, an Extended StackNet approach was proposed. This method expands upon the StackNet concept by incorporating an additional parallel fully connected layer. This extra layer accepts previous angle values as input to achieve even greater accuracy. Four variations of Extended StackNet were tested: EStackNet, EStackNet3D, MiniEStackNet, and MiniEStackNet16. These methods produced even smaller mean absolute errors than the conventional StackNet and CNN-LSTM approaches, with the best result being achieved by MiniEStackNet using a sequence length of four images as input and an error of 0.476 degrees. The Extended StackNet methods also demonstrated more consistent results across different sequence length inputs. These findings suggest that incorporating additional information in the form of previous angles can improve the accuracy of the proposed methods.

Bibliography

[1]Agency, I. E., “Global ev outlook 2020: Entering the decade of electric drive?”, https://www.iea.org/reports/global-ev-outlook-2020, 2020.

[2]Stan čin, H., Mikulčić, H., Wang, X., Duić, N., “A review on alternative fuels in future energy system”, Renewable and Sustainable Energy Reviews, Vol. 128, 10992, aug 2020.

[3]Shahir, V., Jawahar, C., Suresh, P., “Comparative study of diesel and biodiesel on ci engine with emphasis to emissions—a review”, Renewable and Sustainable Energy Reviews, Vol. 45, 2015, str. 686–697.

[4]Jones, D. P., Watkins, A. P., “Droplet size and velocity distributions for spray modelling”, Journal of Computational Physics, Vol. 231, No. 2, Jan. 2012, str. 676–692, dostupno na: https://www.sciencedirect.com/science/article/pii/S0021999111005894

[5]Wu, D., Wang, W., Pang, Z., Cao, S., Yan, J., “Experimental Investigation Of Spray Characteristics Of Diesel-methanol-water Emulsion”, Atomization and Sprays, Vol. 25, No. 8, 2015, str. 675–694.

[6]Eagle, W. E., Morris, S. B., Wooldridge, M. S., “High-speed imaging of transient diesel spray behavior during high pressure injection of a multi-hole fuel injector”, Fuel, Vol. 116, jan 2014, str. 299–309.

[7]Sajjad, H., Masjuki, H. H., Varman, M., Kalam, M., Arbab, M., Imtenan, S., Rahman, S. A., “Engine combustion, performance and emission characteristics of gas to liquid (gtl) fuels and its blends with diesel and bio-diesel”, Renewable and Sustainable Energy Reviews, Vol. 30, 2014, str. 961–986.

[8]Otsu, N., “A Threshold Selection Method from Gray-Level Histograms”, IEEE Transactions on Systems, Man, and Cybernetics, Vol. 9, No. 1, Jan. 1979, str. 62–66.

[9]Ruiz-Rodriguez, I., Pos, R., Megaritis, T., Ganippa, L. C., “Investigation of Spray Angle Measurement Techniques”, IEEE Access, Vol. 7, 2019, str. 22 276–22 289, conference Name: IEEE Access.

[10]Parrish, S. E., Zink, R. J., “Development and application of imaging system to evaluate liquid and vapor envelopes of multi-hole gasoline fuel injector sprays under engine-like conditions”, Atomization and Sprays, Vol. 22, No. 8, 2012, str. 647–661.

[11]Kapusta, Ł. J., “LIF/Mie Droplet Sizing of Water Sprays from SCR System Injector using Structured Illumination”, in Proceedings ILASS–Europe 2017. 28th Conference on Liquid Atomization and Spray Systems, No. September. Valencia: Universitat Politècnica València, sep 2017, str. 6–8, dostupno na: http://ocs.editorial.upv.es/index.php/ILASS/ILASS2017/paper/view/5031

[12]Goodfellow, I., Bengio, Y., Courville, A., Deep learning. MIT press, 2016.

[13]Shen, D., Wu, G., Suk, H.-I., “Deep learning in medical image analysis”, Annual review of biomedical engineering, Vol. 19, 2017, str. 221–248.

[14]Wang, D., Chen, J., “Supervised speech separation based on deep learning: An overview”, IEEE/ACM Transactions on Audio, Speech, and Language Processing, Vol. 26, No. 10, 2018, str. 1702–1726.

[15]Krizhevsky, A., Sutskever, I., Hinton, G. E., “Imagenet classification with deep convolutional neural networks”, Advances in neural information processing systems, Vol. 25, 2012, str. 1097–1105.

[16]LeCun, Y., “The mnist database of handwritten digits”, http://yann. lecun. com/exdb/mnist/, 1998.

[17]Nair, V., Hinton, G. E., “Rectified linear units improve restricted boltzmann machines”, in Icml, 2010.

[18]Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L., “Imagenet: A large-scale hierarchical image database”, in 2009 IEEE conference on computer vision and pattern recognition. Ieee, 2009, str. 248–255.

[19]Everingham, M., Eslami, S. A., Van Gool, L., Williams, C. K., Winn, J., Zisserman, A., “The pascal visual object classes challenge: A retrospective”, International journal of computer vision, Vol. 111, No. 1, 2015, str. 98–136.

[20]Geiger, A., Lenz, P., Urtasun, R., “Are we ready for autonomous driving? the kitti vision benchmark suite”, in 2012 IEEE conference on computer vision and pattern recognition. IEEE, 2012, str. 3354–3361.

[21]Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B., “The cityscapes dataset for semantic urban scene understanding”,

in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, str. 3213–3223.

[22]Carneiro, T., Da Nóbrega, R. V. M., Nepomuceno, T., Bian, G.-B., De Albuquerque, V. H. C., Reboucas Filho, P. P., "Performance analysis of google colaboratory as a tool for accelerating deep learning applications", IEEE Access, Vol. 6, 2018, str. 61 677–61 685.

[23]Huang, Y., Cheng, Y., Bapna, A., Firat, O., Chen, D., Chen, M., Lee, H., Ngiam, J., Le, Q. V., Wu, Y. *et al.*, "Gpipe: Efficient training of giant neural networks using pipeline parallelism", Advances in neural information processing systems, Vol. 32, 2019, str. 103–112.

[24]Oquab, M., Bottou, L., Laptev, I., Sivic, J., "Is object localization for free?-weakly-supervised learning with convolutional neural networks", in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, str. 685–694.

[25]Hung, W.-C., Tsai, Y.-H., Liou, Y.-T., Lin, Y.-Y., Yang, M.-H., "Adversarial learning for semi-supervised semantic segmentation", arXiv preprint arXiv:1802.07934, 2018.

[26]Veit, A., Alldrin, N., Chechik, G., Krasin, I., Gupta, A., Belongie, S., "Learning from noisy large-scale datasets with minimal supervision", in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, str. 839–847.

[27]Wang, P., Chen, P., Yuan, Y., Liu, D., Huang, Z., Hou, X., Cottrell, G., "Understanding convolution for semantic segmentation", in 2018 IEEE winter conference on applications of computer vision (WACV). IEEE, 2018, str. 1451–1460.

[28]He, K., Zhang, X., Ren, S., Sun, J., "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition", 2014, Vol. 8691, str. 346–361, arXiv:1406.4729 [cs], dostupno na: http://arxiv.org/abs/1406.4729

[29]Desai, M., Shah, M., "An anatomization on breast cancer detection and diagnosis employing multi-layer perceptron neural network (mlp) and convolutional neural network (cnn)", Clinical eHealth, 2020.

[30]Simard, P. Y., Steinkraus, D., Platt, J. C. *et al.*, "Best practices for convolutional neural networks applied to visual document analysis.", in Icdar, Vol. 3, No. 2003, 2003.

[31]Çiçek, Ö., Abdulkadir, A., Lienkamp, S. S., Brox, T., Ronneberger, O., "3d u-net: learning dense volumetric segmentation from sparse annotation", in International conference on medical image computing and computer-assisted intervention. Springer, 2016, str. 424–432.

[32]Albawi, S., Mohammed, T. A., Al-Zawi, S., "Understanding of a convolutional neural network", in 2017 International Conference on Engineering and Technology (ICET). Ieee, 2017, str. 1–6.

[33]Kim, P., "Convolutional neural network", in MATLAB deep learning. Springer, 2017, str. 121–147.

[34]Kim, D., Park, S. S., Bae, C., "Schlieren, Shadowgraph, Mie-scattering visualization of diesel and gasoline sprays in high pressure/high temperature chamber under GDCI engine low load condition", International Journal of Automotive Technology, Vol. 19, No. 1, Feb. 2018, str. 1–8, dostupno na: http://link.springer.com/10.1007/s12239-018-0001-8

[35]Payri, R., Salvador, F., Bracho, G., Viera, A., "Differences between single and double-pass schlieren imaging on diesel vapor spray characteristics", Applied Thermal Engineering, Vol. 125, Oct. 2017, str. 220–231, dostupno na: https://linkinghub.elsevier.com/retrieve/pii/S135943111731743X

[36]Lind, S., Retzer, U., Will, S., Zigan, L., "Investigation of mixture formation in a diesel spray by tracer-based laser-induced fluorescence using 1-methylnaphthalene", Proceedings of the Combustion Institute, Vol. 36, Jul. 2016.

[37]Chong, C. T., Hochgreb, S., "Measurements of laminar flame speeds of liquid fuels: Jet-A1, diesel, palm methyl esters and blends using particle imaging velocimetry (PIV)", Proceedings of the Combustion Institute, Vol. 33, Dec. 2011, str. 979–986.

[38]Gibou, F., Hyde, D., Fedkiw, R., "Sharp interface approaches and deep learning techniques for multiphase flows", Journal of Computational Physics, Vol. 380, Mar. 2019, str. 442–463, dostupno na: https://linkinghub.elsevier.com/retrieve/pii/S0021999118303371

[39]Zhang, A., Montanaro, A., Allocca, L., Naber, J., Lee, S. Y., "Measurement of Diesel Spray Formation and Combustion upon Different Nozzle Geometry using Hybrid Imaging Technique", SAE International Journal of Engines, Vol. 7, No. 2, 2014, str. 1034–1043.

[40]Carter, D. W., Hassaini, R., Eshraghi, J., Vlachos, P., Coletti, F., "Multi-scale imaging of upward liquid spray in the far-field region", International Journal of Multiphase Flow, Vol. 132, nov 2020.

[41]Özlüoymak, Ö. B., Bolat, A., "Development and assessment of a novel imaging software for optimizing the spray parameters on water-sensitive papers", Computers and Electronics in Agriculture, Vol. 168, jan 2020.

[42]Payri, R., Salvador, F. J., Martí-Aldaraví, P., Vaquerizo, D., "ECN Spray G external spray visualization and spray collapse description through penetration and morphology analysis", Applied Thermal Engineering, Vol. 112, 2017, str. 304–316, dostupno na: http://dx.doi.org/10.1016/j.applthermaleng.2016.10.023

[43]Rubio-Gómez, G., Martínez-Martínez, S., Rua-Mojica, L. F., Gómez-Gordo, P., De La Garza, O. A., "Automatic macroscopic characterization of diesel sprays by means of a new image processing algorithm", Measurement Science and Technology, Vol. 29, No. 5, 2018.

[44]Bottega, Dongiovanni, "Diesel Spray Macroscopic Parameter Estimation Using a Synthetic Shapes Database", Applied Sciences, Vol. 9, No. 23, dec 2019, str. 5248, dostupno na: https://www.mdpi.com/2076-3417/9/23/5248

[45]Borujeni, A. T., Lane, N. M., Thompson, K., Tyagi, M., "Effects of image resolution and numerical resolution on computed permeability of consolidated packing using LB and FEM pore-scale simulations", Computers and Fluids, Vol. 88, 2013, str. 753–763.

[46]Farhadian, N., Behin, J., Parvareh, A., "Residence time distribution in an internal loop airlift reactor: CFD simulation versus digital image processing measurement", Computers and Fluids, Vol. 167, 2018, str. 221–228.

[47]Mo, J., Tang, C., Li, J., Guan, L., Huang, Z., "Experimental investigation on the effect of n-butanol blending on spray characteristics of soybean biodiesel in a common-rail fuel injection system", Fuel, Vol. 182, 2016, str. 391–401.

[48]Hiroyasu, H., Arai, M., "Structures of fuel sprays in diesel engines", SAE transactions, 1990, str. 1050–1061.

[49]Payri, F., Bermúdez, V., Payri, R., Salvador, F., "The influence of cavitation on the internal flow and the spray characteristics in diesel injection nozzles", Fuel, Vol. 83, No. 4-5, 2004, str. 419–431.

[50]Naruemon, I., Liu, L., Liu, D., Ma, X., Nishida, K., "An Analysis on the Effects of the Fuel Injection Rate Shape of the Diesel Spray Mixing Process Using a Numerical Simulation", Applied Sciences, Vol. 10, No. 14, Jul. 2020, str. 4983.

[51]Zhang, A., Montanaro, A., Allocca, L., Naber, J., Lee, S.-Y., "Measurement of Diesel Spray Formation and Combustion upon Different Nozzle Geometry using Hybrid Imaging Technique", SAE International Journal of Engines, Vol. 7, No. 2, Apr. 2014, str. 1034–1043.

[52]Naber, J. D., Siebers, D. L., “Effects of gas density and vaporization on penetration and dispersion of diesel sprays”, SAE transactions, 1996, str. 82–111.

[53]Kang, J., Bae, C., Lee, K. O., “Initial development of non-evaporating diesel sprays in common-rail injection systems”, International Journal of Engine Research, Vol. 4, No. 4, Aug. 2003, str. 283–298.

[54]Payri, R., Gimeno, J., Bracho, G., Vaquerizo, D., “Study of liquid and vapor phase behavior on Diesel sprays for heavy duty engine nozzles”, Applied Thermal Engineering, Vol. 107, Aug. 2016, str. 365–378.

[55]Pastor, J. V., Arrègle, J., Palomares, A., “Diesel spray image segmentation with a likelihood ratio test”, Applied Optics, Vol. 40, No. 17, Jun. 2001, str. 2876.

[56]Minaee, S., Boykov, Y. Y., Porikli, F., Plaza, A. J., Kehtarnavaz, N., Terzopoulos, D., “Image Segmentation Using Deep Learning: A Survey”, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021, str. 1–1, conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.

[57]Sorensen, T. A., “A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on danish commons”, Biol. Skar., Vol. 5, 1948, str. 1–34.

[58]Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L., “ImageNet: A large-scale hierarchical image database”, in 2009 IEEE Conference on Computer Vision and Pattern Recognition, Jun. 2009, str. 248–255, iSSN: 1063-6919.

[59]Smith, L. N., “Cyclical Learning Rates for Training Neural Networks”, in 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), Mar. 2017, str. 464–472.

[60]Dhanachandra, N., Manglem, K., Chanu, Y. J., “Image Segmentation Using K -means Clustering Algorithm and Subtractive Clustering Algorithm”, Procedia Computer Science, Vol. 54, 2015, str. 764–771.

[61]Feng, D., Wenkang, S., Liangzhou, C., Yong, D., Zhenfu, Z., “Infrared image segmentation with 2-D maximum entropy method based on particle swarm optimization (PSO)”, Pattern Recognition Letters, Vol. 26, No. 5, Apr. 2005, str. 597–603.

[62]Leung, C.-K., Lam, F.-K., “Image segmentation using maximum entropy method”, in Proceedings of ICSIPNN ’94. International Conference on Speech, Image Processing and Neural Networks, Apr. 1994, str. 29–32 vol.1.

[63]Pickett, L. M., Manin, J., Payri, R., Bardi, M., Gimeno, J., "Transient Rate of Injection Effects on Spray Development", Sep. 2013, str. 2013–24–0001.

[64]Eagle, W. E., Malbec, L.-M., Musculus, M. P., "Measurements of Liquid Length, Vapor Penetration, Ignition Delay, and Flame Lift-Off Length for the Engine Combustion Network 'Spray B' in a 2.34 L Heavy-Duty Optical Diesel Engine", SAE International Journal of Engines, Vol. 9, No. 2, Apr. 2016, str. 910–931.

[65]Lin, T.-Y., Dollar, P., Girshick, R., He, K., Hariharan, B., Belongie, S., "Feature Pyramid Networks for Object Detection", in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Honolulu, HI: IEEE, Jul. 2017, str. 936–944.

[66]Chaurasia, A., Culurciello, E., "LinkNet: Exploiting Encoder Representations for Efficient Semantic Segmentation", 2017 IEEE Visual Communications and Image Processing (VCIP), Dec. 2017, str. 1–4, arXiv: 1707.03718.

[67]Ronneberger, O., Fischer, P., Brox, T., "U-net: Convolutional networks for biomedical image segmentation", in International Conference on Medical image computing and computer-assisted intervention. Springer, 2015, str. 234–241.

[68]Chen, Y., Li, J., Xiao, H., Jin, X., Yan, S., Feng, J., "Dual Path Networks", dostupno na: http://arxiv.org/abs/1707.01629 ArXiv:1707.01629 [cs] version: 2. Jul. 2017.

[69]Tan, M., Le, Q. V., "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks", dostupno na: http://arxiv.org/abs/1905.11946 ArXiv:1905.11946 [cs, stat]. Sep. 2020.

[70]Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.-C., "MobileNetV2: Inverted Residuals and Linear Bottlenecks", dostupno na: http://arxiv.org/abs/1801.04381 ArXiv:1801.04381 [cs]. Mar. 2019.

[71]He, K., Zhang, X., Ren, S., Sun, J., "Deep Residual Learning for Image Recognition", dostupno na: http://arxiv.org/abs/1512.03385 ArXiv:1512.03385 [cs]. Dec. 2015.

[72]Simonyan, K., Zisserman, A., "Very Deep Convolutional Networks for Large-Scale Image Recognition", dostupno na: http://arxiv.org/abs/1409.1556 ArXiv:1409.1556 [cs]. Apr. 2015.

[73]van Dyk, D. A., Meng, X.-L., "The Art of Data Augmentation", Journal of Computational and Graphical Statistics, Vol. 10, No. 1, Mar. 2001, str. 1–50, dostupno na: http://www.tandfonline.com/doi/abs/10.1198/10618600152418584

[74]Simonyan, K., Zisserman, A., “Very Deep Convolutional Networks for Large-Scale Image Recognition”, dostupno na: http://arxiv.org/abs/1409.1556 ArXiv:1409.1556 [cs]. Apr. 2015.

[75]Tan, M., Le, Q. V., “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks”, dostupno na: http://arxiv.org/abs/1905.11946 ArXiv:1905.11946 [cs, stat]. Sep. 2020.

[76]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”, dostupno na: http://arxiv.org/abs/1905.02244 ArXiv:1905.02244 [cs] version: 5. Nov. 2019.

[77]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”, dostupno na: http://arxiv.org/abs/1704.04861 ArXiv:1704.04861 [cs]. Apr. 2017.

[78]Rumelhart, D. E., Hinton, G. E., Williams, R. J., “Learning representations by back-propagating errors”, Nature, Vol. 323, No. 6088, 1986, str. 533–536.

[79]Donahue, J., Hendricks, L. A., Rohrbach, M., Venugopalan, S., Guadarrama, S., Saenko, K., Darrell, T., “Long-term Recurrent Convolutional Networks for Visual Recognition and Description”, dostupno na: http://arxiv.org/abs/1411.4389 ArXiv:1411.4389 [cs]. May 2016.

[80]Smith, L. N., “A disciplined approach to neural network hyper-parameters: Part 1 – learning rate, batch size, momentum, and weight decay”, dostupno na: http://arxiv.org/abs/1803.09820 ArXiv:1803.09820 [cs, stat]. Apr. 2018.

List of Figures

List of Tables

Biography

Fran Huzjan, born in 1996 in Zagreb, Croatia, is a computer scientist with a strong background in deep learning and image analysis. Fran began his academic journey at the High School of Tituš Brezovački in Zagreb, before pursuing his undergraduate studies in computer science at the University of Zagreb's Faculty of Electrical Engineering and Computing. He graduated with a Bachelor's degree in 2018, following the completion of his thesis, "Optimization of GPS Attack Parameters with Evolutionary Algorithm." Continuing his education at the same institution, Fran obtained his Master's degree in 2020, with his master's thesis focusing on "Parking Space Occupancy Detection." Concurrently, from September 2018 to August 2020, he gained practical experience as a machine learning intern at 3MI Lab, where he led a student team in utilizing deep learning methods for parking space occupancy detection. In October 2020, Fran commenced his Doctoral Studies in computer science at the University of Zagreb's Faculty of Electrical Engineering and Computing, under the guidance of Professor Sven Lončarić, Ph.D. His current research explores spray image analysis using deep learning techniques. As part of the RESIN project, Fran collaborates with the Faculty of Mechanical Engineering and Naval Architecture at the University of Zagreb as a young researcher. Fran's academic responsibilities also include serving as a teaching assistant for the courses "Deep Learning" and "Scripting Languages." His research interests encompass computer vision, deep learning, machine learning, image analysis, and image processing. Fran has also been involved in organizing several workshops and summer schools, and he has authored a Q2 journal publication and a conference paper.

List of publications

Journal papers

1. **Huzjan, F.**, Jurić, F., Lončarić, S., & Vujanović, M. (2023). Deep Learning-based Image Analysis Method for Estimation of Macroscopic Spray Parameters. Neural Computing Applications 35, 9535–9548

Conference papers

1. **Huzjan, F.**, Jurić, F., Vujanović, M., Lončarić, S.: Deep learning-based cone angle estimation using spray sequence images. In: Proceedings of the 2023 8th International Conference on Machine Learning Technologies. ICMLT '23, pp. 208–213.

Životopis

Fran Huzjan, rođen 1996. godine u Zagrebu, Hrvatska, računarni je znanstvenik s jakom pozadinom u dubokom učenju i analizi slika. Fran je započeo svoje akademsko putovanje u Gimnaziji Tituša Brezovačkog u Zagrebu, prije nego što je nastavio preddiplomski studij računarstva na Fakultetu elektrotehnike i računarstva, Sveučilišta u Zagrebu. Stekao je titulu prvostupnika računarstva 2018. godine nakon završetka svog preddiplomskog rada pod nazivom "Optimizacija parametara GPS napada uz pomoć evolucijskog algoritma". Nastavljajući svoje obrazovanje na istoj ustanovi, Fran je stekao magisterij 2020. godine, s temom diplomskog rada "Detekcija zauzeća parkirnih mjesta". U isto vrijeme, od rujna 2018. do kolovoza 2020. godine, stekao je praktično iskustvo kao pripravnik u području strojnog učenja u 3MI Labu, gdje je vodio studentski tim u primjeni metoda dubokog učenja za detekciju zauzeća parkirnih mjesta. U listopadu 2020. godine, Fran je započeo doktorski studij računarske znanosti na Fakultetu elektrotehnike i računarstva, Sveučilišta u Zagrebu, pod mentorstvom profesora Svena Lončarića, gdje se bavi analizom slika spreja koristeći metode dubokog učenja. Kao dio RESIN projekta, Fran surađuje s Fakultetom strojarstva i brodogradnje, Sveučilišta u Zagrebu kao mladi istraživač. Franove akademske odgovornosti također uključuju obavljanje poslova asistenta na kolegijima "Duboko učenje" i "Skriptni jezici". Njegovi istraživački interesi obuhvaćaju računalni vid, duboko učenje, strojno učenje, analiza i obrada slika. Fran je također sudjelovao u organizaciji nekoliko radionica i ljetnih škola te je autor publikacije u Q2 časopisu i rada na konferenciji.