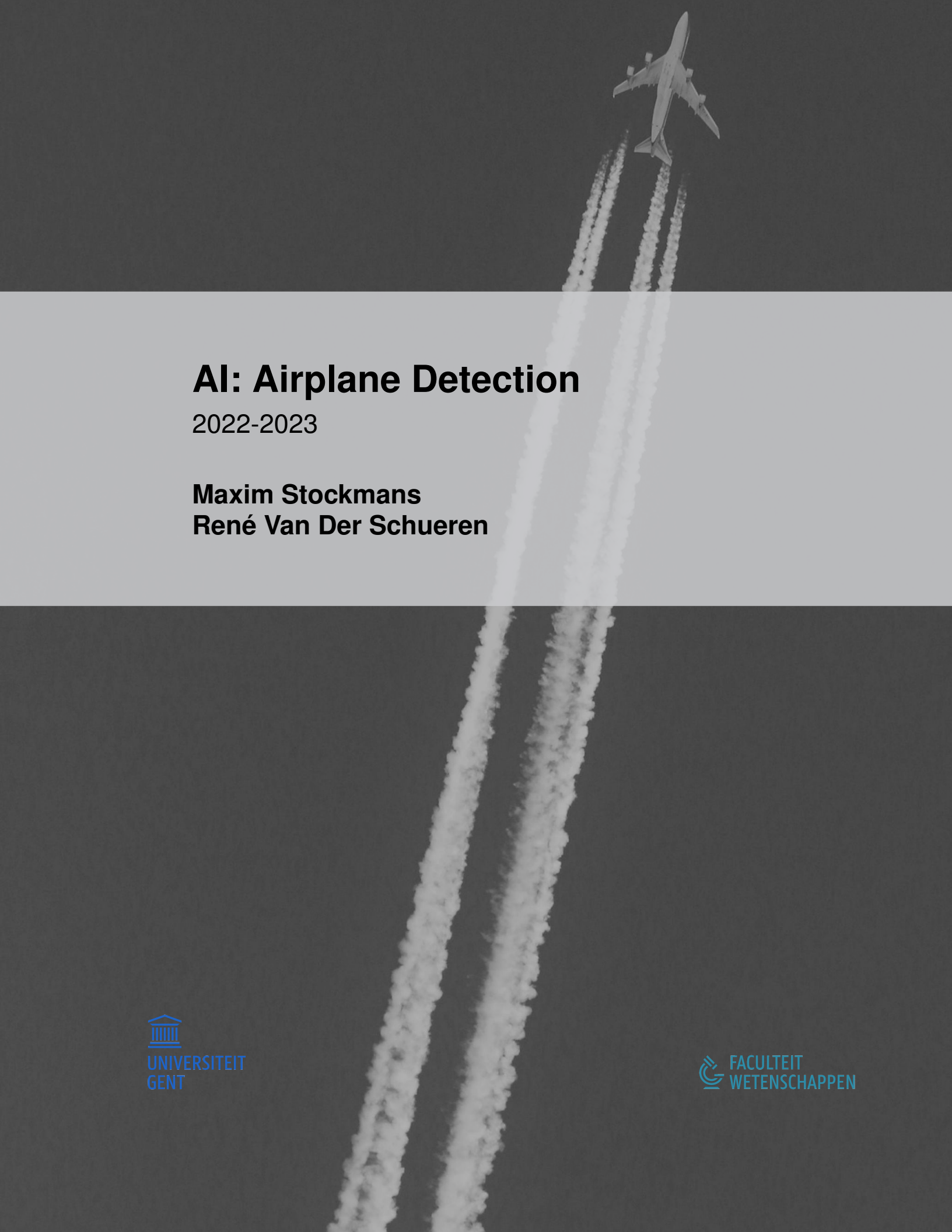


AI: Airplane Detection

2022-2023

Maxim Stockmans
René Van Der Schueren

UNIVERSITEIT GENT

FACULTEIT WETENSCHAPPEN

Inhoudsopgave

1 Inleiding

1.1 Introductie

In de wereld van de luchtvaart bestaan er ontelbaar veel verschillende modellen van vliegtuigen. Het is voor een persoon onmogelijk om elk van deze modellen te herkennen. In dit onderzoek zullen we dit probleem trachten op te lossen met artificiële intelligentie (AI) gedreven afbeeldingsherkenning. Er zijn verschillende toepassingen waar gebruik kan worden gemaakt van automatische herkenning van vliegtuigmodellen:

- Vliegtuigspotters, zij spotten en nemen foto's van vliegtuigen als hobby en willen graag het exacte model kunnen identificeren.
- Luchtverkeersleiding kan betere keuzes nemen indien ze bij visueel contact het exacte vliegtuigmodel kunnen identificeren.
- Militaire instanties kunnen een tactisch voordeel hebben indien ze het exacte model van een vliegtuig kunnen herkennen. Verder kunnen er ook burgerslachtoffers voorkomen worden door misidentificatie van verkeersvliegtuigen tegen te gaan.

Eerst zullen we het probleem zo precies mogelijk omschrijven, alsook de gebruikte dataset. Vervolgens bespreken we hoe we het resultaat hebben verbeterd. Hiervoor pasten we technieken zoals data modificatie toe, waar we de input aanpassen om zo een beter resultaat te bekomen. We hebben ook data augmentatie technieken toegepast, deze vergroten de hoeveelheid trainingsdata aan de hand van verschillende technieken zonder de originele dataset te vergroten. Hierna bespreken we enkele modellen voor onze AI die een opmerkelijk resultaat gaven.

1.1.1 Beschrijving van het probleem

De hoeveelheid vliegtuigmodellen die bestaat is zeer groot. We willen dus een model trainen om op basis van een foto het juiste model te herkennen.

(a) Ilyushin Il-76

(b) General Dynamics F-16

Figuur 1.1: Twee voorbeelden van vliegtuigen.

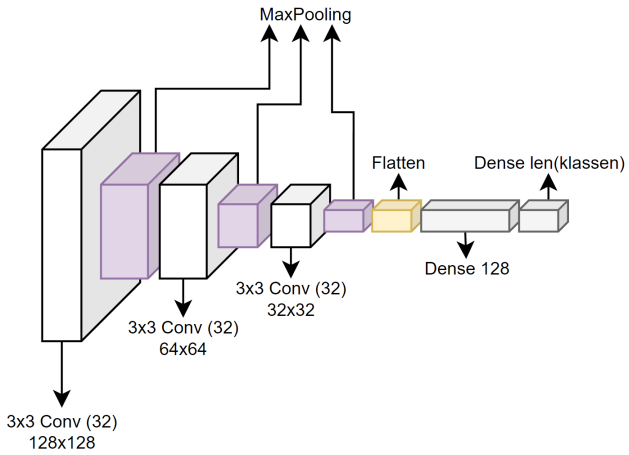
Figuur 1.1 toont een voorbeeld van twee vliegtuigen die de AI zou moeten kunnen herkennen. Verder wensen we dat indien de AI een fout maakt, deze een vliegtuig herkent dat in dezelfde categorie ligt. Een passagiersvliegtuig verwarren met een ander passagiersvliegtuig kan een bruikbaar resultaat geven. Een passagiersvliegtuig verwarren met een militair vliegtuig echter niet.

1.2 Mogelijke oplossingsmethode

Voor dit onderzoek kozen we voor een ResNet model. ResNets zijn een type CNN die effectief zijn voor afbeeldingsherkenningstaken. Ze gebruiken een "residuele blok" -architectuur die helpt om zeer diepe netwerken gemakkelijker te trainen door gradiënten gemakkelijker door het netwerk te laten stromen. ResNets zijn succesvol geweest in een groot aantal afbeeldingsherkenningstaken [3].

1.2.1 Het initiële model

Voor het initiële model hebben we een simpel ResNet gebruikt, gebaseerd op *Practicum 3: Deep learning* [6] en uit de *Image classification: Tensorflow Core* [5] documentatie.



Figuur 1.2: Het initieel ResNet model.

Op figuur 1.2 kunnen we alle lagen van het ResNet model zien. Het model bevat drie convolutie lagen met elk 32 filters, waartussen steeds een *pooling* laag zit. Dit model hebben we doorheen het onderzoek gebruikt als basis om, aan de hand van experimenten, het resultaat te verbeteren.

1.3 Beschrijving van de data

1.3.1 TorchVision Dataset

Initieel werd geëxperimenteerd met de dataset FGVCAIRCRAFT [2], dit is een dataset die standaard aanwezig is in de PyTorch bibliotheek. Deze bevat 100 verschillende vliegtuigmodellen met elk 100 verschillende foto's. Deze dataset bleek relatief klein. Het initiële

model haalde hier ook maar een lage accuraatheid op. We besloten om zelf een grotere dataset samen te stellen.

1.3.2 Dataset jetphotos.com

Jetphotos.com is een website waar vliegtuigspotters hun zelfgenomen foto's uploaden. Ze duiden vervolgens ook de specifieke naam van het model aan. Dit vormt dus de perfecte dataset. De website bevat meer dan vijf miljoen foto's van vliegtuigen. Wij scrapten uiteindelijk 67.286 foto's van 5.318 verschillende vliegtuigmodellen. Elke foto had een resolutie van 400x225. De dataset bedroeg in het totaal 3.25GB en werd verder doorheen het onderzoek gebruikt.

2 Methodiek

Om het resultaat van ons model te verbeteren zijn we eerst aan de slag gegaan met onze input data. Zo hebben we geëxperimenteerd met het filteren en groeperen van de dataset, het omzetten van afbeeldingen naar grijstinten en de resolutie te herschalen. Ook hebben we data augmentatie technieken uitgeprobeerd om zo uit onze bestaande dataset meer data punten te genereren, en zo overfitting tegen gaan.

2.1 Data modificatie

2.1.1 Groeperen en filteren

Een initiële run op onze dataset gaf een accuraatheid van 30%. Deze lage accuraatheid was te verklaren door verschillende problemen met de dataset. Als eerste probleem bevatte de dataset te veel klassen. De modelnamen van de vliegtuigen waren zeer specifiek, hierdoor leken veel klassen zeer goed op elkaar.

(a) Airbus A319-114

(b) Airbus A319-115

Figuur 2.1: Voorbeeld van twee gelijkaardige vliegtuigen.

Hierdoor besloten we bepaalde modellen te groeperen aan de hand van een set van handmatig geschreven regels. Zo werden de voorbeelden uit figuur 2.1 gecombineerd tot een klasse 'Airbus A319'.

Verder zijn er veel zeldzamere vliegtuigen waarvan er weinig foto's in de dataset aanwezig waren. Om dit op te lossen filterden we alle klassen die minder dan een bepaald aantal foto's bevatten.

Als we een minimum van 100 foto's toepasten per klasse, kwamen we op een resultaat van 52 verschillende klassen met een totaal van 44.255 foto's. Op een resolutie van 128x128 bekwamen we een accuraatheid van 70%.

We probeerden dit ook met een minimum van 500 foto's, de accuraatheid steeg dan ongeveer 5%. We verloren hier een groot aantal klassen voor een kleine verbetering, daarom werken we in dit onderzoek met een minimum van 100 foto's per klasse.

2.1.2 Resolutie

Eén van de belangrijkste factoren waarmee we experimenteerden was de resolutie waarop we ons AI model trainde en valideerde. Ons doel was om een balans te vinden tussen de tijd die we nodig hadden om het model te trainen en het eindresultaat.

(a) 32x32 pixels

(b) 256x256 pixels

Figuur 2.2: Het verschil tussen een lage en hoge resolutie.

Om dit te doen voerden we een reeks aan experimenten uit waar de resolutie varieerde van 32x32 pixels tot 256x256 pixels. We zijn begonnen met de laagste resolutie en verhoogden dit geleidelijk aan om zo te onderzoeken of het verschil in accuraatheid de verhoogde uitvoeringstijd kon verantwoorden.

Figuur 2.3: De uiteindelijke resolutie, 128x128 pixels.

Zoals verwacht bracht een hogere resolutie een beter eindresultaat en een hogere trainingstijd per epoch. Voor dit onderzoek kozen we 128x128 pixels als de gulden middenweg tussen uitvoeringstijd en eindresultaat. We zagen hier toch nog een 5% boost in accuraatheid tegenover 64x64 pixels. Met een uitvoeringstijd die zeker aanvaardbaar was om verdere experimenten op onze machine uit te voeren.

Tabel 2.1: De machine.

Cores	8
Threads	16
Klokfrequentie	4.7 GHz
GPU	GTX 1070
RAM	32 GB

2.1.3 Afbeeldingen in grijstinten

Ook hebben we geëxperimenteerd met het omzetten van onze afbeeldingen zodat deze enkel grijstinten bevatten. We verwerken dan geen extra kleurinformatie, waardoor we een pixel door één enkele waarde kunnen voorstellen. Hiervoor waren dit er drie, één voor elk kleurkanaal (rood, groen en blauw). Een voordeel dat hieruit volgt is dus dat er minder computerkracht nodig is om de afbeeldingen te verwerken in vergelijking met afbeeldingen die kleur bevatten. Aangezien we lokaal trainden met een grote dataset leek het dus zeker een goed pad om te onderzoeken.

Afbeeldingen met enkel grijstinten hebben ook enkele nadelen. We verliezen data die mogelijks het resultaat van ons model kan verbeteren om het specifieke vliegtuig model te herkennen. Aangezien kleur soms een belangrijke eigenschap kan zijn van bepaalde vliegtuigmodellen, bijvoorbeeld markeringen of kleurenschema's bij militaire vliegtuigen.

(a) Kleur (b) Grayscale

Figuur 2.4: Verlies van informatie bij afbeeldingen met een laag contrast.

Ook kan het licht in de foto het resultaat beinvloeden wanneer we afbeeldingen in grijstinten gebruiken. Bijvoorbeeld wanneer het licht oneven is verdeeld of wanneer het contrast laag is, kunnen bepaalde details wegvallen of minder duidelijk zichtbaar worden in vergelijking met de originele foto.

Dit verklaart ook de resultaten die we uit deze experimenten gehaald hebben. Onze accuraatheid met het originele model op de ongefilterde dataset daalde namelijk sterk. Hiervoor behaalden we zo een 30% accuraatheid en met onze trainingsdata in grijstinten was dit slechts 18%. Hieruit concludeerden we dus dat we best de kleurinformatie kunnen behouden. Dit hebben we in volgende experimenten dan ook steeds gedaan.

2.2 Data augmentatie

Doorheen de verschillende experimenten werd telkens duidelijke overfitting waargenomen. Hier gaat het model dus steeds meer convergeren naar zijn trainingsdataset en niet meer of zelfs slechter worden in de validatie dataset. Om dit tegen te gaan pasten we data augmentatie toe. Hier gaan we de bestaande trainingsdata lichtjes aanpassen om zo de hoeveelheid input te vergroten. Door deze telkens aan te passen kan de AI zich niet vasthouden aan eigenschappen van de dataset.

2.2.1 Gerandomiseerde rotatie

Als eerste oplossing voor overfitting proberen we een gerandomiseerde rotatietechniek. Dit zorgt er voor dat de trainingsinput telkens aan een lichtjes andere hoek wordt ingegeven. Hierdoor wordt het model getraind om de vliegtuigen te kunnen herkennen onder verschillende hoeken [5].

In de experimenten namen we echter een negatief effect waar, zelfs bij een maximale rotatie van 72 graden. Dit kunnen we verklaren door twee effecten. Ten eerste zal er veel informatie verloren raken bij het roteren van afbeeldingen met een lage resolutie. De meeste foto's in onze dataset zijn vanaf de zijkant genomen. De rotatie van het vliegtuig is dus de hoek die zijn neus met de grond maakt, ook bekend als de invalshoek. De Boeing 737, het meest voorkomende vliegtuig ter wereld, en ook het meest voorkomende in onze dataset, zal in normale omstandigheden nooit een grotere invalshoek maken met de grond dan 16 graden. De meeste vliegtuigen zullen zich met gemiddeld weinig rotatie in de dataset bevinden. De uitdaging voor de AI word dus aanzienlijk groter als we deze rotatie invoegen [7].

We zagen na het toepassen van gerandomiseerde rotatie een verlies van minstens 5% accuraatheid, dit werd niet verder onderzocht.

2.2.2 Gerandomiseerde horizontale spiegeling

Door een gerandomiseerde horizontale spiegeling uit te voeren op de trainingsdata leren we ons model het vliegtuig in twee richtingen te herkennen. Meestal zijn vliegtuigen symmetrisch. Horizontale spiegeling van een foto heeft dus geen effect op welke klasse het is. We willen er dus ook voor zorgen dat ons model hier geen rekening mee houdt, gezien dit ook in echte data zal verschillen van onze trainingsdata [5].

Bij het toepassen van dit effect zagen we een verbetering van 11% accuraatheid tegenover ons initieel model.

2.2.3 Dropout rate

Dropout rate is een regularisatietechniek die overfitting bij het trainen tegengaat. Het zorgt ervoor dat sommige neuronen gedeactiveerd worden tijdens het trainen. Zo moet het model redundante voorstellingen van de data leren om zo de data te generaliseren [5].

Doordat we een dropout rate gebruikten motiveerden we ons model om de specifieke eigenschappen van de vliegtuigen te leren die minder beïnvloedbaar waren door ruis. We hebben geëxperimenteerd met de dropout rate, om zo tot de conclusie te komen dat een dropout rate van 50% de beste resultaten gaf. Dit zorgde voor een goede balans tussen regularisatie en performantie. Bij een hogere dropout rate was er te veel regularisatie. Hierdoor kon het mogelijks belangrijke eigenschappen van vliegtuigen niet langer herkennen en dus een slechter resultaat geven. Bij een lagere dropout rate hadden we dan overfitting. Een dropout rate van 50% was dus een goede afweging.

We zagen in onze experimenten een verbetering van ongeveer 6% accuraatheid.

2.2.4 Gerandomiseerde zoom

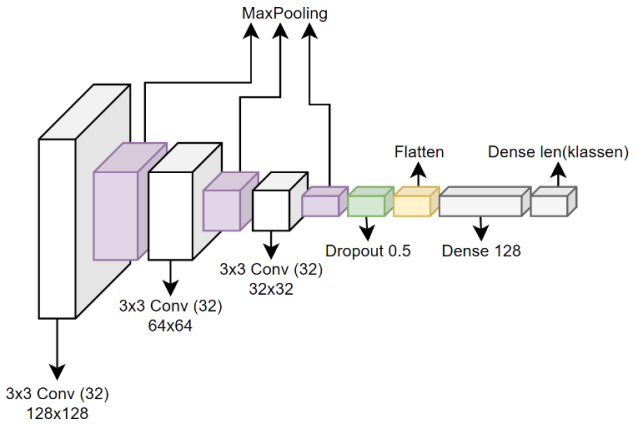
Het gerandomiseerd in- of uitzoomen op de trainingsdata had dan weer een positief effect. Dit zorgt ervoor dat ons model de data kan herkennen op verschillende schalen. Dit omdat de grootte van het vliegtuig op de foto niet specifiek is voor een bepaald model, dit hangt af vanwaar de foto genomen wordt. Door de gerandomiseerde zoom toe te voegen proberen we dus te vermijden dat ons model hierop traint.

We bereikten het beste resultaat bij een zoom tussen -10% en 20%. We zagen een verbetering van ongeveer 2% accuraatheid.

2.3 Verbeterde modellen

2.3.1 Gebruikte lagen in het ResNet

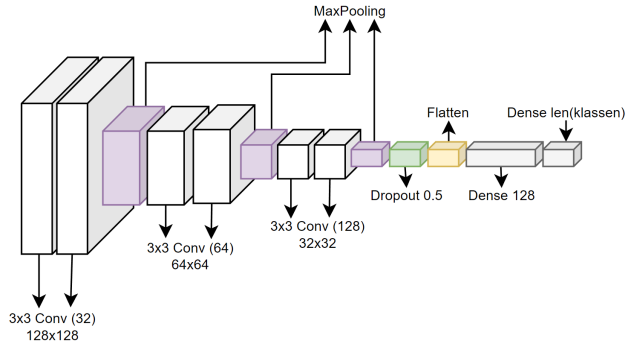
Figuur 2.5 toont de staat van het model na het toevoegen van de dropout. Met de optimale data augmentatie en modificatie bereiken we met dit model een accuraatheid 83,93%.



Figuur 2.5: Verbeterd origineel model.

Het is belangrijk om de, door het model gebruikte, soort en aantal lagen zorgvuldig te selecteren. Zo benutten we de mogelijkheden van de ResNet-architectuur volledig. Na het vinden van de optimale data augmentatie en modificatie zochten we ook een beter model. Hieronder bespreken we de modellen die merkbare resultaten gaven.

2.3.2 Een groter model

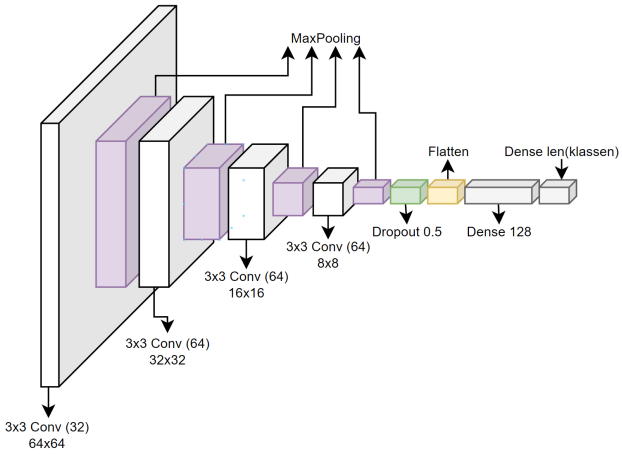


Figuur 2.6: Een groter model.

Het eerste wat onderzocht werd is de verbetering in accuraatheid indien we in het model meer convolutielagen gebruiken met elk meer filters. Figuur 2.6 toont het grotere model waar we op eindigden. Dit model bevat diepere convolutielagen, en 2 convolutie lagen per *pooling* laag.

Dit grotere model is computationeel zwaarder om te trainen. De trainingstijd steeg van ongeveer 20 seconden per epoch voor het initiële model naar ongeveer 2 minuten per epoch voor dit model. De accuraatheid steeg tegenover het vorige model naar 84,52%. Dit is een relatief kleine verbetering tegenover het veel snellere originele model.

2.3.3 Het finale model



Figuur 2.7: Het finale model.

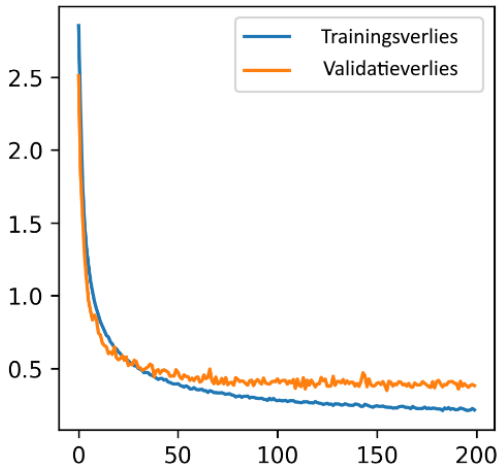
Figuur 2.7 toont het finale model. Hier besloten we om een laag toe te voegen met 32 filters en met een eigen *pooling* laag. Het aantal filters van de andere lagen werd verhoogd naar 64. Dit bleek de beste balans te zijn tussen de tijd per epoch, de hoeveelheid overfitting en het uiteindelijke resultaat. We zagen hier een grote verbetering in precisie tegenover het vorige model.

3 Resultaten

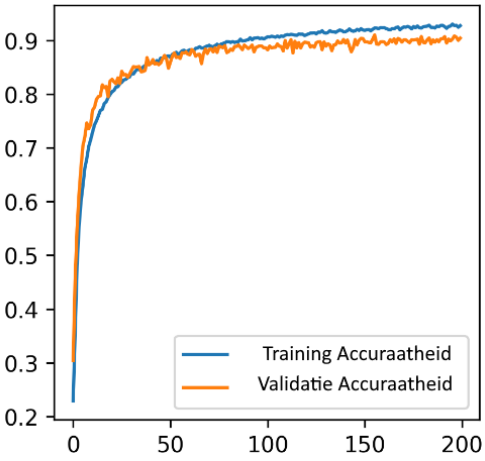
Ons beste eindresultaat bekwamen we na het finale model te trainen voor 156 epochs. De algemene accuraatheid die we hierdoor bekwamen was 91.01%. Door alle toegepaste data augmentatie en modificatie kunnen we nu efficiënt trainen zonder extreme overfitting.

3.1 Accuraatheid & loss

In figuur 3.2 zien we dat de verbetering in de accuraatheid van het model tijdens het trainen stagneert na een 150-tal epochs. Op figuur 3.1 zien we wel dat het validatieverlies boven het trainingsverlies ligt na slechts 50 epochs, er is dus zeker sprake van overfitting. Maar de validatie en trainings accuraatheid blijven dalen tot rond de 150ste epoch.



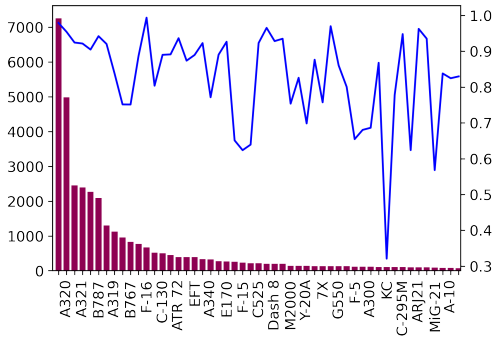
Figuur 3.1: Trainings- en validatieverlies in functie van de epoch.



Figuur 3.2: Trainings- en validatie accuraatheid in functie van de epoch.

3.2 Gebalanceerde accuraatheid

Onze dataset is een reflectie van de echte wereld. Vliegtuigen die veel voorkomen in de realiteit komen ook veel voor in onze dataset. Het is dus belangrijk om de gebalanceerde accuraatheid te bekijken. Dit is het gemiddelde van de accuraatheid per klasse. We bekomen dan nog steeds een hoge accuraatheid, namelijk 82.6%.



Figuur 3.3: De linkse as en balken tonen de hoeveelheid foto's per klasse, de rechtse as en lijn tonen de gebalanceerde accuraatheid per klasse. (Niet alle labels zijn getoond.)

Figuur 3.3 plot de hoeveelheid foto's en de accuraatheid per klasse. We zien hier dat klassen met een lagere hoeveelheid foto's meer kans hebben om een lagere accuraatheid te hebben. Toch kunnen we niet spreken van een lineair verband aangezien er een aanzienlijk aantal van de klassen met weinig foto's toch nog goed presteert. Veel van de vliegtuigmodellen met weinig foto's zijn meestal ook modellen met speciale kenmerken. Hierdoor kan de AI uit een lagere hoeveelheid foto's meer leren. Veel van de passagiersvliegtuigen lijken goed op elkaar, maar deze komen ook veel voor in onze dataset. Onze dataset zou dus wel winst kunnen halen uit meer foto's van sommige zeldzame vliegtuigen. Maar er is echter wel goed gebalanceerd in het voordeel van de in realiteit veel voorkomende klassen en moeilijk te onderscheiden klassen.

3.3 Verwarringsmatrix

Appendix A toont de verwarringsmatrix. Hier toont de y-as de klasse waarvan een foto aan het model gegeven werd. De x-as toont welke klasse dit model teruggaf voor deze foto. De klassen werden op elke as gesorteerd van veel foto's in de dataset naar weinig foto's in de dataset. Elke horizontale lijn, dus alle resultaten voor de foto's van een klasse, werden genormaliseerd.

Onze hoge gebalanceerde accuraatheid wordt gereflecteerd door de diagonaal. Dit zijn allemaal juist geclassificeerde foto's, namelijk 91.01% van de foto's. We zien ook dat de meest voorkomende vliegtuigen, zowel in het echt als in onze dataset, het vaakst gekozen worden als het model de foute keuze neemt. Dit kunnen we zien in de eerste kolommen van de verwarringsmatrix.

Er kunnen enkele oorzaken zijn waarom ons model dit doet. Zo heeft hij mogelijks de kenmerken en patronen van deze twee vliegtuigen het meest geleerd omdat deze vaker voorkomen in de dataset. Ook zou het kunnen dat ons model gewoon een willekeurige keuze neemt wanneer hij niet zeker is en omdat deze vliegtuigen meer voorkomen in de dataset ook vaker gekozen worden.

Ook zien we verwarring bij de vliegtuigen waar minder foto's voor zijn (de onderste rijen van de verwarringsmatrix). Dit zijn vooral kleine en militaire vliegtuigen. Mogelijks kunnen we op deze klassen dus ook nog een beter resultaat behalen indien we voor deze modellen meer data verzamelen.

De grootste fout zien we links onderaan in de grafiek. Dit is de klasse KC, met de laagste accuraatheid die we ook in figuur 3.3 opmerkten. Dit is te verklaren doordat er nauwelijks verschil tussen deze twee vliegtuigmodellen zit. Het is namelijk de passagiers- en militaire variant van de Airbus 330. Figuur 3.4 toont de twee vliegtuigen met hun weinige verschil.

(a) Airbus A330 (passagiersvliegtuig)

(b) Airbus KC (militaire variant)

Figuur 3.4: Vergelijking Airbus 330 en Airbus KC.

Veel van de andere klassen met lage accuraatheid kunnen op gelijkaardige manier verklaard worden.

4 Conclusie

Dit onderzoek had als doel om de effectiviteit te onderzoeken van AI gedreven afbeeldingsherkenning om met een hoge zekerheid het specifieke model van vliegtuigen voor een gegeven afbeelding te raden. We hebben in totaal meer dan 35 sets aan hyperparameters onderzocht en meer dan 1400 epochs getraind, om zo ons optimaal eindresultaat te bekomen. Hieruit kunnen we besluiten dat afbeeldingsherkenning een goede techniek is voor het classificeren van vliegtuigmodellen, en dat dit een goede fundering is om verder op te bouwen.

Om verder te gaan in dit onderzoek kunnen we een uitgebreidere dataset verzamelen met meer foto's van meer klassen. Ook zouden we de impact van meer computerkracht kunnen onderzoeken. Momenteel zijn alle experimenten op een relatief lichte machine uitgevoerd en werden we hierdoor gelimiteerd in de hoeveelheid invoer en de complexiteit van het gebruikte ResNet model. Indien deze limitering wegvalt kunnen we mogelijks onze experimenten uitvoeren met een hogere resolutie, meer en complexere lagen en op een grotere dataset.

Referenties

[1] *Deep Learning for Aircraft Recognition Part I: Building a Convolutional Neural Network (CNN) from Scratch.* URL: https://chrischow.github.io/dataandstuff/2020-09-05-deep-learning-for-aircraft-recognition/.

[2] *FGVCAIRCRAFT.* URL: https://pytorch.org/vision/main/generated/torchvision.datasets.FGVCAircraft.html.

[3] Kaiming He e.a. *Deep Residual Learning for Image Recognition.* 2015. DOI: 10.48550/ARXIV.1512.03385. URL: https://arxiv.org/abs/1512.03385.

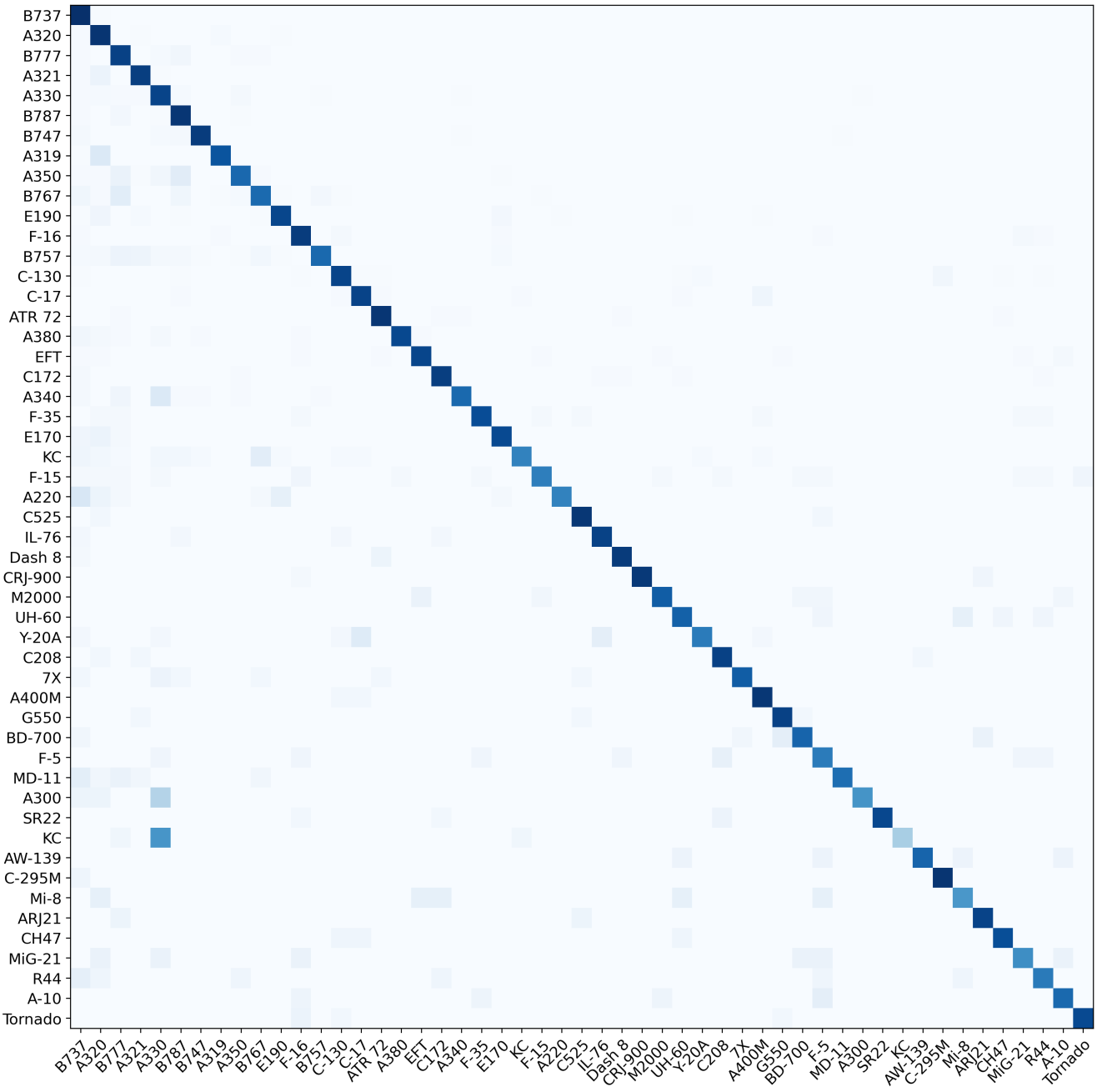
[4] *Hyperparameters Tuning in Practice: Pandas vs. Caviar.* URL: https://www.coursera.org/lecture/deep-neural-network/hyperparameters-tuning-in-practice-pandas-vs-caviar-DHNcc.

[5] *Image classification: Tensorflow Core.* URL: https://www.tensorflow.org/tutorials/images/classification.

[6] Yvan Saeys. *Practicum 3: Deep learning.* 2022.

[7] *What is the takeoff pitch for a Boeing 737-800.* URL: https://community.infiniteflight.com/t/what-is-the-takeoff-pitch-for-a-boeing-737-800/394028.

5 Appendix



Appendix A: Verwarringsmatrix van het eindresultaat.