Faculteit Industriële Ingenieurswetenschappen

master in de industriële wetenschappen: elektromechanica

Masterthesis

Computervisiesysteem voor kwaliteitscontrole van blauwe natuursteen op basis van een convolutioneel neuraal netwerk

Michiel Claes
Dries Deridder
Scriptie ingediend tot het behalen van de graad van master in de industriële wetenschappen: elektromechanica

PROMOTOR :
Prof. dr. ir. Johan BAETEN

PROMOTOR :
Dhr. Jan VAN ELSACKER

Gezamenlijke opleiding UHasselt en KU Leuven

UHASSELT KU LEUVEN

Universiteit Hasselt | Campus Diepenbeek | Faculteit Industriële Ingenieurswetenschappen | Agoralaan Gebouw H - Gebouw B | BE 3590 Diepenbeek

Universiteit Hasselt | Campus Diepenbeek | Agoralaan Gebouw D | BE 3590 Diepenbeek
Universiteit Hasselt | Campus Hasselt | Martelarenlaan 42 | BE 3500 Hasselt

UHASSELT KU LEUVEN

2022
2023

Faculteit Industriële Ingenieurswetenschappen

master in de industriële wetenschappen: elektromechanica

Masterthesis

Computervisiesysteem voor kwaliteitscontrole van blauwe natuursteen op basis van een convolutioneel neuraal netwerk

Michiel Claes
Dries Deridder

Scriptie ingediend tot het behalen van de graad van master in de industriële wetenschappen: elektromechanica

PROMOTOR :

Prof. dr. ir. Johan BAETEN

PROMOTOR :

Dhr. Jan VAN ELSACKER

UHASSELT KU LEUVEN

Woord vooraf

Graag presenteren we u onze masterproef. Deze masterproef is de laatste stap in het behalen van onze opleiding elektromechanica met focus op automatisering binnen de industriële ingenieurswetenschappen aan de UHasselt en KU Leuven. Deze stap was een uitdagende maar leerrijke ervaring.

De uitdaging binnen deze masterproef lag hem in het uitwerken van het convolutioneel netwerk en het maken van de data die nodig is om een convolutioneel netwerk te laten trainen. Het verzamelen en verwerken van de data heeft meer tijd in beslag genomen dan eerst verwacht waardoor er minder tijd in andere delen van deze masterproef zijn kunnen gaan.

Deze masterproef konden we niet alleen verwezenlijken, daarom willen we graag volgende mensen bedanken. Allereerst willen we onze externe promotor, Dhr. Jan Van Elsacker bedanken voor het aanleveren van dit onderwerp en de praktijkgerichte visie. Vervolgens willen we ook graag onze interne promotor, prof. dr. ir. Johan Baeten, bedanken voor de goede opvolging en alle adviezen die we van hem mochten ontvangen.

Graag willen we ook Massimo Giardina bedanken omdat wij zijn server mochten gebruiken voor het meermaals trainen van het convolutioneel netwerk. Tot slot willen we ook graag onze familie en vrienden bedanken van wie we extra raad en hulp mochten ontvangen.

Wij wensen u veel leesplezier toe.

Inhoudsopgave

Tabellenlijst

Figuurlijst

Abstract

Optidrive te Leuven is gespecialiseerd in het bouwen van robotoplossingen voor natuursteen verwerkende bedrijven. Het doel van deze masterproef is het ontwikkelen van een software omgeving waarmee een kwaliteitscontrole kan uitgevoerd worden om zo een operator te helpen een gekende soort natuursteen in kwaliteitsklassen in te delen. Het bedrijf stelt een opstelling ter beschikking waarop zowel trainings- als testbeelden gemaakt kunnen worden.

Bij het ontwikkelen van dit programma zijn er twee vereisten. Ten eerste moet de exacte locatie van de natuurstenen op de transportband bepaald kunnen worden. Deze wordt bepaald op het moment dat de natuursteen zich onder de camera van de opstelling bevindt. De tweede en belangrijkste taak van het programma is het analyseren van de steenoppervlakte en zo defecten zoals fossielen, witte of zwarte vlekken in de stenen te detecteren.

Er werden zowel beelden gemaakt van stenen op een transportband als de afzonderlijke oppervlakken, zodat het ontworpen convolutioneel netwerk voldoende data heeft om zichzelf te trainen. De lokalisatie van de stenen gebeurt door middel van randdetectie. Voor de detectie van defecten in de stenen wordt er gebruik gemaakt van een convolutioneel neuraal netwerk. De stenen worden opgedeeld in deelsegmenten. Elk segment wordt onderzocht naar defecten. Met behulp van een wegingsfactor, verschillend per defect, krijgt elke steen een score die de kwaliteit van de steen aangeeft. Op basis hiervan maakt een operator de keuze in welke kwaliteitsklasse de steen behoort.

Abstract in English

Optidrive, a company located in Leuven is specialized in building robot solutions for stone processing companies. The main task of this master's thesis is to develop a software program that can perform a quality check that can help an operator to divide a given natural stone into different quality classes. On site there is a machine present that can be used for making training and test data.

There are two requirements when developing this program. The first task is to determine the exact location of the natural stones on the conveyor belt. The location is determined when they are moved under a camera on the machine. The second and most important task of the program is to analyze the stone surface and identify defects in the stones such as fossils and white or black spots.

Images were made of the stones on a conveyor belt as well as the individual stone surfaces, this to make sure the developed Convolutional Network has enough data to train itself. The localization of the stones is done by means of edge recognition. For the detection of defects in the stones, a convolutional neural network is used. The stones are divided into sub-segments. Each segment is then checked for these defects. A weighting factor is introduced depending on the importance of the defect, each stone is given a score that indicates the quality of the stone. Based on this score the operator places the stone in a certain quality class.

1 Inleiding

1.1 Situering

Deze masterproef, getiteld 'Computervisiesysteem voor kwaliteitscontrole van blauwe natuursteen op basis van een convolutioneel neuraal netwerk', wordt mogelijk gemaakt door de firma Optidrive te Leuven. Optidrive is een automatisatiebedrijf dat bewerkingsmachines maakt voor de steenverwerkende nijverheid. Deze machines kunnen ieder een verschillende bewerking realiseren. Zo zijn er bewerkingsmachines die stenen producten zagen, frezen, schuren of polijsten, … Deze machines maken gebruik van robots. De robots vormen de fysieke link tussen de verschillende bewerkingsmachines. De Optidrive-robots geven ook verwerkte informatie door over elk specifiek product of elke te bewerken steen. Optidrive hecht veel belang aan het correct verwerken van deze informatie. Dit zodat deze robots robuust en universeel toepasbaar zijn. Een tweede belangrijke pijler voor Optidrive is het eenvoudig bedienen van deze robots [1].

Optidrive maakt gebruik van een basisrobot. Deze basisrobot bestaat uit een robotarm en de sturing van de robotarm. De robotarm laat toe, met behulp van een gemonteerde grijper, objecten te verplaatsen in de ruimte. Afhankelijk van de grijper kan de robot verschillende functies hebben. De implementaties van Optidrive maken gebruik van een grijper uitgerust met zuignappen. De robotarm wordt uitgebreid met verschillende sensoren en camera's. Optidrive werkt aan de visuele analyses voor het lokaliseren van de stenen alsook het implementeren van de bijhorende software. Het is dan ook zeer belangrijk voor het bedrijf dat de overstap naar deze Optidrive-implementaties en -software zeer eenvoudig is voor de steenverwerkende bedrijven.

In het academiejaar 21'-22' werd dit onderwerp reeds een eerste keer opgenomen. De focus lag toen vooral bij het detecteren van fouten gebruikmakend van andere technieken. Deze masterproef wordt dit jaar 22'-23' dus voor de tweede keer opgenomen. In deze Masterproef wordt er verder gegaan op de detectie van de fouten op basis van een convolutioneel netwerk en wordt er gewerkt aan een implementatie van de kwaliteitsevaluaties van bewerkte stenen. Deze kwaliteitsevaluaties omvatten een visuele analyse van de bewerkte stenen met de mogelijkheid om via deze analyse de stenen te categoriseren in verschillende kwaliteitsklassen.

De huidige opstelling bestaat uit een transportband en een robotarm, uitgerust met een camera en geschikte manipulator voor het verplaatsen van stenen. Deze opstelling kan de stenen lokaliseren en vastgrijpen om ze op een pallet te leggen.

1.2 Probleemstelling

In heel wat steenverwerkende bedrijven wordt er momenteel een manuele controle gedaan op onzuiverheden, beschadigingen of andere defecten. Om deze manuele controle te kunnen automatiseren is Optidrive op zoek naar een manier om verwerkte stenen te kunnen classificeren in verschillende kwaliteitsklassen.

Er werd al een machine ontworpen die de stenen kan categoriseren op basis van de grootte van deze stenen. Dit is echter niet voldoende. Stenen kunnen namelijk ook variëren in kleur, dikte, kwaliteit en fouten in de afwerking. De kleur in eenzelfde steen kan verschillend zijn. Er kunnen ongewenste

donkere of witte lijnen, maar ook fossielen in de stenen aanwezig zijn. Dit is nadelig voor de kwaliteitsklasse van de stenen. Tijdens de bewerkingen van de stenen kunnen er fouten optreden. Door een plaatselijk defect van de stenen kunnen er tijdens het zagen, frezen of polijsten hoekjes afbreken van de stenen. Als deze bewerkingsfouten te groot zijn moet een steen dus afgekeurd worden.

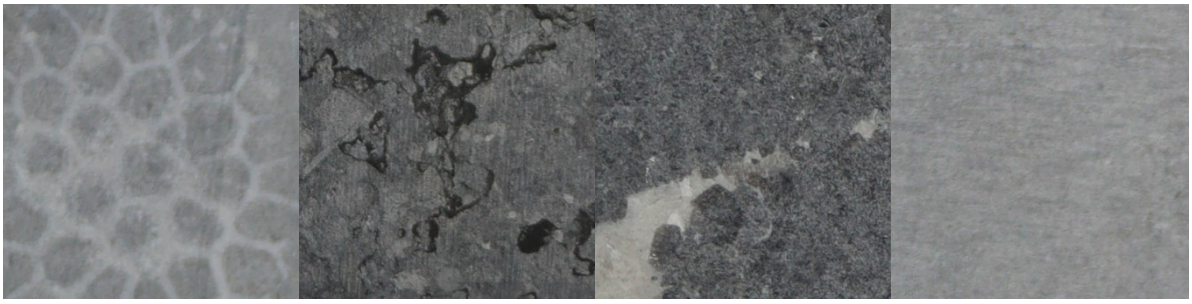
1.3 Doelstellingen

Zoals eerder vermeld is het hoofddoel van deze masterproef het controleren van natuurstenen met behulp van camera's en deze stenen vervolgens in te delen in de verschillende kwaliteitsklassen.

Vooraleer er kan begonnen worden met het classificeren van de stenen moet er eerst en vooral een opstelling gebouwd worden waarop de detectie van de stenen kan gebeuren. Hiervoor is er een bestaande opstelling, bestaande uit twee transportbanden, ter beschikking gesteld. Een van de vereisten voor de te bouwen opstelling is dat ze makkelijk te verwijderen of makkelijk op- en afbreekbaar is. De bestaande machine moet immers buiten deze masterproefopdracht ook zijn functie als testopstelling voor verschillende proeven kunnen behouden.

Nadat de opstelling gebouwd is zijn er twee functies die de controle-eenheid moet kunnen uitvoeren. Eerst en vooral moet deze in staat zijn om de stenen te lokaliseren op de transportband. In tegenstelling tot vorig academiejaar waarbij er één soort steen gebruikt werd dient de machine dit jaar de mogelijkheid te bezitten om stenen te detecteren van verschillende afmetingen met verschillende oppervlaktebewerkingen.

De tweede functie is het detecteren van onzuiverheden in de stenen, wat de belangrijkste functie is voor het bepalen van de kwaliteitsklasse van de steen. Aangezien stenen uit de natuur gehaald en niet in een fabriek geproduceerd worden, is de kans op onzuiverheden vrij groot. De vaakst voorkomende visuele fouten in de te controleren stenen zijn kleurverschillen, fossielen en witte lijnen. Het detecteren van onzuiverheden houdt in dat het softwareprogramma in staat is elk steenoppervlak te analyseren, en deze vervolgens toe te kennen aan één van de vijf categorieën. De eerste en meest voorkomende categorie is "Goed" deze wordt toegekend wanneer er geen defecten aanwezig zijn. Vervolgens is er voor elk defect een categorie, "Zwart", "Wit" en "Fossiel". De vijfde en laatste categorie genaamd "Moeilijk" wordt toegekend wanneer het onduidelijk is in welke categorie de steen behoort of indien er meer dan één defect aanwezig is. Figuur 1 geeft een voorbeeld van de beschreven categorieën.



Figuur 1: Een voorbeeld van de verschillende foutklassen waarbij het oppervlak in respectievelijke volgorde behoort tot de categorieën: Fossielen, Zwart, Wit, Goed

Zodra het programma de steen beoordeeld heeft en toegekend aan een bepaalde categorie, wordt deze categorie aan een operator weergegeven. De hoeveelheid fouten in de steen wordt weergegeven door een getal, berekend uit de som van wegingsfactoren per aanwezige fout. Dit getal helpt de operator bij de finale beslissing tot welke kwaliteitsklasse de steen behoort.

1.4 Materiaal en methode

Vooraleer er effectief begonnen kan worden met het bouwen, programmeren, … dient er eerst een literatuurstudie te gebeuren om extra informatie te verzamelen over mogelijke technieken die gebruikt kunnen worden om de probleemstelling op te lossen. Hierbij moet er gekeken worden naar de huidige technieken voor computervisie, beeldverwerking, machine learning, AI, … Deze technieken moeten dan vergeleken worden om te bepalen welke het meest van toepassing zijn voor de gegeven opdracht en doelstelling.

Zodra er genoeg informatie verzameld is (om duidelijke keuzes te maken waarom bepaalde technieken gekozen worden) kan er begonnen worden aan het schetsen, ontwerpen en opbouwen van de opstelling.

Zoals eerder vermeld wordt deze masterproef dit jaar voor de tweede keer uitgevoerd. De gebruikte materialen voor deze opstelling zoals camera's, lasers, …mogen hergebruikt worden. '1.3 Doelstellingen' vermeldt dat er dit academiejaar ook een bestaande machine ter beschikking gesteld wordt. Optidrive heeft voldoende materialen ter beschikking gesteld zodat de aanpassingen aan de opstelling mogelijk zijn. Als er een aankoop nodig is kan dit op voorhand besproken worden.

Nadat de opstelling opgebouwd is, is het de bedoeling zoveel mogelijk data te verzamelen in de vorm van beeldmateriaal. Aangezien er meer afmetingen toegelaten moeten worden dan vorig jaar is het ook mogelijk om meer beelden te maken.

Er kan al gewerkt worden met de beelden die vorig jaar gemaakt zijn. Deze zijn niet van de beste kwaliteit en zijn dus enkel nuttig als aanzet naar de effectieve beelden. Door eerder in het proces te starten aan de visuele kwalificatie is het mogelijk om hier meer tijd aan te besteden.

Voor de effectieve verwerking van de gegevens is gevraagd in open CV te werken. De keuze van de programmeertaal waarin dit gebeurt, ligt niet vast. Doorheen het project zal vergeleken worden welke programmeertaal (Python, C++ of C #) het best geschikt is voor deze toepassing.

1.5 Gerelateerd werk

Aangezien het de bedoeling is om stenen te categoriseren in verschillende kwaliteitsklassen, is het belangrijk te weten of er fouten aanwezig zijn, welke fouten dit zijn en waar deze zich bevinden. Voordat er via beeldverwerking op een foto fouten in stenen kunnen gedetecteerd worden, moet de steen dus eerst gelokaliseerd worden op deze foto. Kortweg, voor zowel de lokalisatie van de steen als het opsporen van fouten zijn er camera's en sensoren nodig. Om de verschillende mogelijke technieken voor zowel de lokalisatie als de kwaliteitscheck te vergelijken zijn er verschillende papers geraadpleegd. De verschillende technieken worden hieronder beschreven.

Uit de bronnenstudie van afgelopen jaar blijkt dat er vooral is gekeken naar het oppervlak van de stenen. Enkele technieken die hieruit voortkwamen waren sheet of light en Lidar. Dit zijn twee technieken die ook bij microscopen gebruikt worden om het reliëf van bepaalde stalen in kaart te brengen. In deze masterproef ligt de focus op het visueel detecteren van de fouten in de stenen.

1.5.1 Segmentatie

Er zijn verschillende manieren om segmentatie uit te voeren. Zo bestaat er vormsegmentatie en randsegmentatie. Zoals [2] aangeeft, is vormsegmentatie vaak een voorbereidende fase op randsegmentatie. In die paper is de werking van vormsegmentatie terug te vinden zoals hierna beschreven. Om de positie van een voorwerp te bepalen wordt gebruik gemaakt van een R-tabel. In een eerdere trainingsfase werd deze tabel opgesteld als zijnde de gemiddelde vorm van het te zoeken voorwerp en de bijhorende afwijkingen in alle richtingen. Met behulp van een Hough transformatie wordt er in een afbeelding gezocht naar punten die overeenstemmen met punten in de R-tabel. Van hieruit wordt met gradiënten gezocht naar de mogelijke aanliggende punten. Op deze manier kan er een ruwe vorm bekomen worden die de locatie van het voorwerp aangeeft.

Dit kan in dit onderzoek handig van pas komen om in eerste instantie na te gaan naar waar de te detecteren steen zich bevindt onder de camera. Zodra de steen gelokaliseerd is kan er met behulp van randsegmentatie de precieze rand bepaald worden. Als deze randen dan vergeleken worden met rechte lijnen, kan de haaksheid van de stenen bepaald worden, en of er eventueel stukjes van een rand afgebroken zijn.

Deze techniek kan ook gebruikt worden voor het detecteren van de onzuiverheden in de stenen zoals fossielen. Een tweede en misschien wel betere optie voor dit onderdeel van de opdracht is het werken met samenvoegen van regio's. Paper [3] beschrijft hoe er met behulp van deze techniek gezocht kan worden naar vlekjes op de huid die een aanwijzing zijn van een bepaald soort kanker. Aangezien het hierbij gaat om vlekjes van allerlei verschillende vormen en maten, stemt dit overeen met de zoektocht naar fossielen en andere onzuiverheden in de stenen in deze thesis. Er wordt beschreven dat de beelden eerst vooraf moeten verwerkt worden voordat ze gecontroleerd worden op vlekjes.

[4] beschrijft uitgebreider de aanpak en werking van het effectief samenvoegen van regio's. Er bestaan hierbij ook weer meerdere manieren waarop de regio's samengevoegd kunnen worden. Enerzijds door groei, hierbij wordt er vertrokken vanuit 1 pixel met een bepaalde intensiteit en wordt er vervolgens naar de omliggende pixels gekeken om de intensiteit te vergelijken. Afhankelijk hiervan wordt de aanliggende pixel bij de vorm gevoegd of bij de omgeving. Een tweede manier is door het beeld continu te blijven opsplitsen tot het opgedeeld is in vlakken met dezelfde intensiteit. De vlakken met gelijke intensiteit worden achteraf terug samengevoegd.

1.5.2 Software

Computervisie:

Computervisie omvat verschillende technieken die het mogelijk maken om nuttige data te halen uit verworven foto's of video's. Deze data kunnen dan gebruikt worden om, in het geval van deze masterproef, stenen te classificeren volgens verschillende kwaliteitsnormen. Om deze classificatie mogelijk te maken worden er eerst beelden van de transportband genomen. Voor deze opnamen

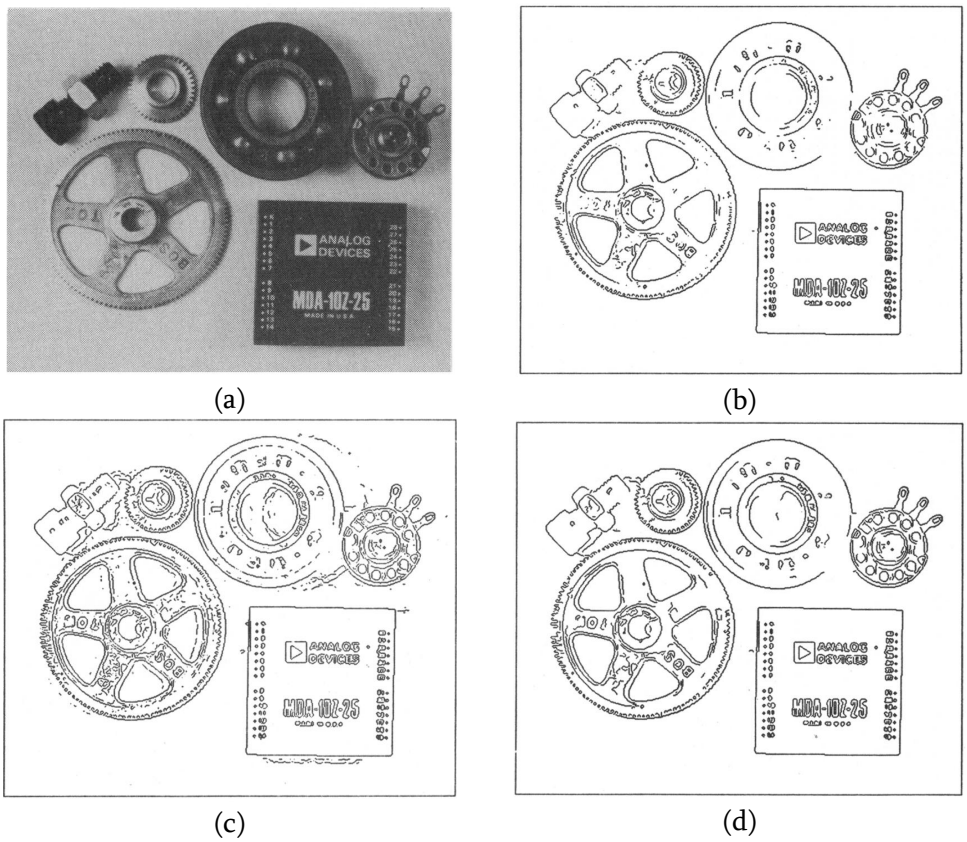
van de beelden wordt gebruikt gemaakt van een kleurencamera van het type Canon EOS 250D. Aangezien hier met een kleurencamera gewerkt wordt bestaan de beelden uit drie verschillende lagen dit volgens het RGB-schema. Iedere laag bestaat uit een matrix van getallen. Deze getallen variëren tussen 0 en 255. Als een waarde laag is, komt dit overeen met het ontbreken van deze kleur en een hoge waarde overeenkomt met het volledig aanwezig zijn van deze kleur. Om gebruik te kunnen maken van computervisie worden deze lagen meestal omgevormd naar een enkele laag met een grijze kleur. Met deze grijze foto is er dan de mogelijkheid om verschillende detecties uit te voeren.

Randdetectie op basis van threshold:

Een eerste detectie is de randdetectie. Deze is nodig zodat de steen gevonden kan worden binnen in het beeld dat gegenereerd wordt door de camera. Een eerste mogelijkheid is het gebruiken van een threshold. Deze methode selecteert de pixels waarvan de grijswaarden binnen een gegeven grens ligt. Binnen de gemaakte opstelling kan de transportband vuil worden en is het dus aangeraden om gebruik te maken van een dynamische grens. Otsu's thresholding [5] is een manier die een dynamische grens legt binnen de threshold, dit is dus voordelig bij veranderingen in de omgeving. Echter zal in deze opstelling niet altijd stenen aanwezig zijn op de transportband. Hierdoor zal de dynamische grens snel foutief worden en is deze manier van werken minder aangeraden.

Randdetectie op basis van Canny:

Een Canny randdetectie maakt gebruik van de verandering van de grijswaarden. Als er op een plaats een plotse verandering van grijswaarden aanwezig is dan is de kans zeer groot dat er zich hier een rand van een object zal bevinden. Hoe gevoelig deze detectie moet zijn kan ingesteld worden via de parameters. Figuur 1 geeft een voorbeeld van deze randdetectie bij verschillende parameters. De eigenschappen van de detectie kunnen aangepast worden met behulp van hysterese. Figuur 2 geeft een voorbeeld van de invloed van de threshold op de uitkomst alsook de invloed van hysterese.

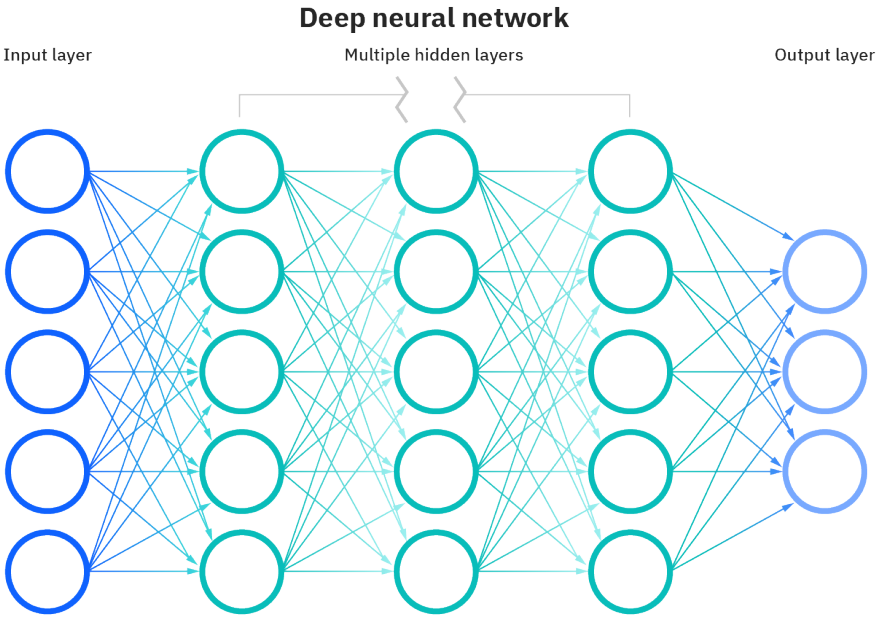


Figuur 2: (a) originele afbeelding. (b) afbeelding met threshold op T_1. (c) afbeelding met threshold op 2 T_1. (d) Afbeelding met threshold met Hysterese gebruikmakend van thresholds van (a) en (b) [6]

Detectie van fouten

In de stenen kunnen fouten aanwezig zijn. Om deze fouten te detecteren zal er gebruik gemaakt worden van een convolutioneel neuraal netwerk. Dit netwerk zal getraind moeten worden op verschillende fouten.

Dit netwerk bestaat uit verschillende ingangen, verborgen lagen en uitgangen. Figuur 3 geeft een voorbeeld van een convolutioneel neuraal netwerk. Deze netwerken maken verschillende berekeningen vertrekkend van de ingangen. Om zinvolle resultaten te bekomen, moet het netwerk eerst getraind worden. Hierin geldt hoe meer data gebruikt wordt voor het trainen van het model hoe beter. Binnen deze masterproef is het mogelijk om dezelfde data meerdere keren te gebruiken voor het trainen van het model. De afbeelding kan geroteerd of gespiegeld worden. Dit zorgt er dus voor dat het model meer data ter beschikking heeft voor de training.



Figuur 3: Schema van convolutioneel neuraal netwerk [7]

2 Methode

2.1 Opstelling

Figuur 4 geeft de gebruikte opstelling weer voor het maken van foto's waarbij de stenen op de transportband liggen. In deze opstelling werd er gebruik gemaakt van een spiegelreflexcamera van het type Canon EOS 250D gecombineerd met een Canon 18-55mm f/3.5-5.6 DC III lens. Deze combinatie laat toe om hoogwaardige foto's te nemen van de stenen, dit zodat ieder detail van de stenen duidelijk zichtbaar is. Bij het vergelijken van deze beelden met de beelden van vorig jaar blijkt dat deze kwalitatief betere beelden zorgen voor een enorme verbetering in de werking van het netwerk.

De camera-instellingen van de Canon EOS 250D in deze opstelling zijn als volgt:

- Diafragma: Het diafragma is ingesteld op een waarde van f/4.5. Door deze diafragmawaarde te selecteren, wordt de scherptediepte beheerst en worden de lichtinval en de scherpte van het beeld geoptimaliseerd.
- Belichtingstijd: De belichtingstijd is ingesteld op 1/15 seconden. Deze specifieke tijdsduur bepaalt de duur waarin het licht op de beeldsensor valt, waardoor de gewenste belichtingsniveaus worden bereikt voor de vastgelegde beelden.
- ISO-waarde: De ISO-instelling is ingesteld op 400. ISO staat voor International Organization for Standardization en verwijst naar de lichtgevoeligheid van de beeldsensor. Door te kiezen voor ISO 400 wordt een optimale balans bereikt tussen de lichtgevoeligheid en het minimaliseren van beeldruis.
- Brandpuntafstand: De brandpuntafstand is vastgelegd op 29 mm. Deze parameter geeft de afstand aan tussen het brandpunt van de lens en de beeldsensor, wat resulteert in een specifiek gezichtsveld en perspectief voor de vastgelegde beelden.

Verder werd er ook gebruik gemaakt van een LED-paneel. Deze zorgt voor een egale belichting van de stenen.

Figuur 4: Gebruikte opstelling voor het maken van foto's waarbij de stenen op de transportband liggen

2.2 Softwareomgeving

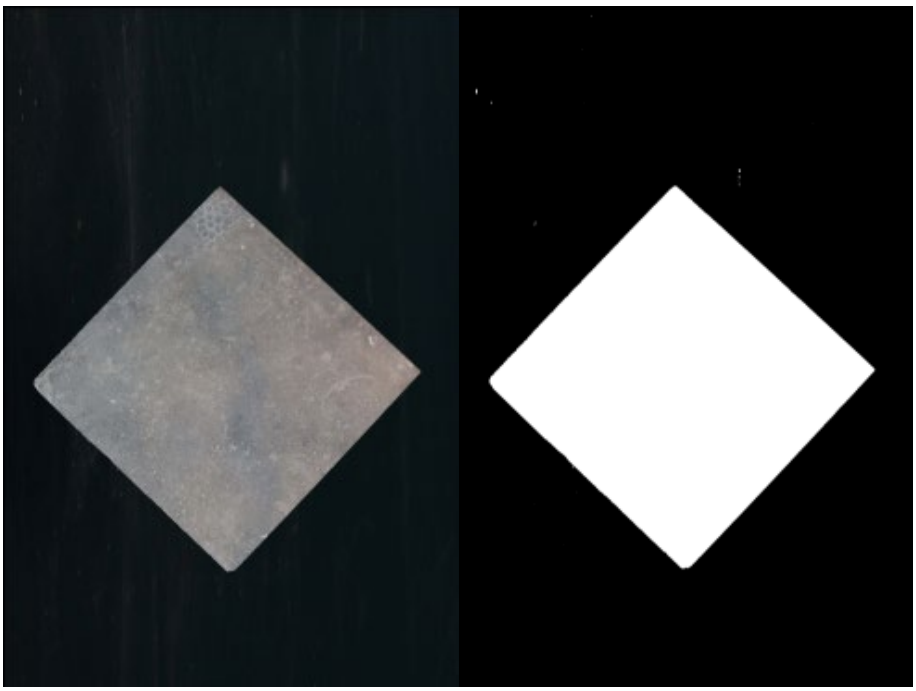
Deze thesis maakt gebruikt van Python 3.10.7. De extra geïnstalleerde bibliotheken met versienummer zijn te vinden in de bijlagen. Voor het ontwikkelen van het programma werd gebruik gemaakt van de Jupyter notebook omgeving [9]. Het uiteindelijke programma maakt gebruik van de Python omgeving.

2.3 Lokalisatie op de transportband

Het lokaliseren van de steen binnen de afbeelding, met als achtergrond de transportband, vormt een belangrijke stap binnen deze thesis. Voor deze opdracht bestaat het lokaliseren uit drie delen, namelijk: het detecteren van de steen binnen de afbeelding, het roteren van deze afbeelding zodat de randen een parallelle positie krijgen met de randen van de afbeelding en als derde het uitsnijden van het oppervlak van de steen.

2.3.1 Detecteren van de steen binnen de afbeelding

Voor de detectie van de steen binnen de afbeelding maakt deze thesis gebruik van de grijswaarde van de originele afbeelding. Vervolgens wordt er een threshold toe gepast op deze grijswaarde. Dit resulteert zich in een afbeelding waar de uitkomsten de waarde 0 of 1 kunnen hebben. De uitkomsten hebben de waarde 0 als de grijswaarden niet binnen de threshold bevindt en worden met een zwarte kleur aangeduid. De uitkomsten hebben de waarde 1 als de grijswaarden zich wel binnen de threshold bevindt en worden met een witte kleur aangeduid. Figuur 5 geeft een voorbeeld van de threshold waarbij de grijswaarden liggen tussen [50, 200].



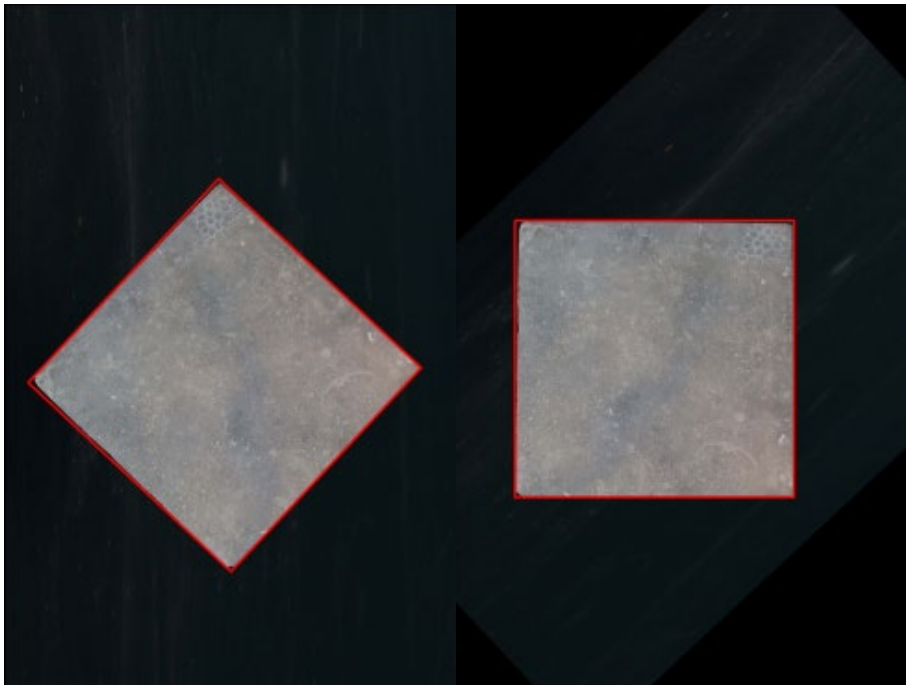
Figuur 5: Steen liggend op een transportband en de overeenkomstige threshold waarbij de grijswaarde ligt tussen [50, 200]. Een witte pixel geeft aan dat de grijswaarde van de originele pixel behoort tot de threshold.

Deze afbeelding wordt vervolgens gebruikt om de grootst mogelijke contour van de witte vlakken te bepalen. Deze contour omsluit dus de gehele steen. Figuur 6 geeft een voorbeeld van een zo groot mogelijke contour bij een steen op de transportband. Deze contour wordt bepaald door de gegenereerde hoekpunten van de berekening. Deze hoekpunten kunnen vervolgens gebruikt worden om de rotatie van de steen binnen de afbeelding te bepalen.

Figuur 6: De grootst mogelijke contour op basis van de threshold, weergeven op de originele afbeelding. De contour wordt weergeven door de rode rechthoek

2.3.2 Roteren van de steen

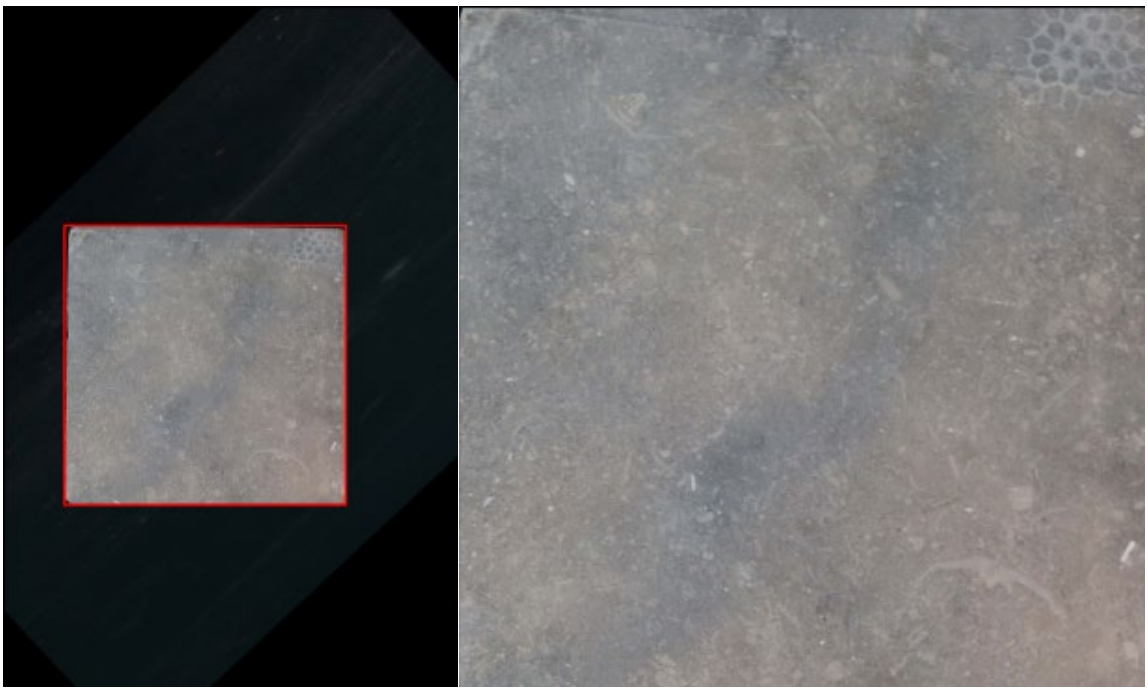
Het convolutioneel netwerk verwacht vierkante oppervlakken voor de berekening. Om ervoor te zorgen dat een zo groot mogelijk gedeelte van het steenoppervlak getest kan worden door het convolutioneel netwerk is het van cruciaal belang dat de steen parallel georiënteerd staat ten opzichte van de rand van de afbeelding. Om de hoek van de steen ten opzichte van de rand van de afbeelding te kunnen bepalen wordt er gebruik gemaakt van de contour van de steen, zoals besproken in 2.3.1 Detecteren van de steen binnen de afbeelding, de hoekpunten van deze contour laten toe om de gewenste hoek te bepalen. Vervolgens wordt een rotatiematrix opgesteld. Deze rotatiematrix wordt vervolgens gebruikt om de afbeelding en de contour te roteren rond het middelpunt van de afbeelding. Figuur 7 geeft een voorbeeld van de rotatie op basis van de contour.



Figuur 7: Rotatie van de afbeelding op basis van de contour. De contour wordt weergeven door de rode rechthoek.

2.3.3 Uitsnijden oppervlak van de steen

Een laatste stap van de lokalisatie is het uitsnijden van het oppervlak van de steen. De uitsnijding maakt gebruik van de reeds bepaalde contour van de steen om het gewenste oppervlak te bepalen. Dit oppervlak wordt verder verkleind op basis van de offset. Deze offset laat toe de rand van de steen te verwijderen zodat enkel het oppervlak overgehouden wordt. Figuur 8 geeft een voorbeeld weer van het uitsnijden van het gewenste oppervlak op basis van de contour.



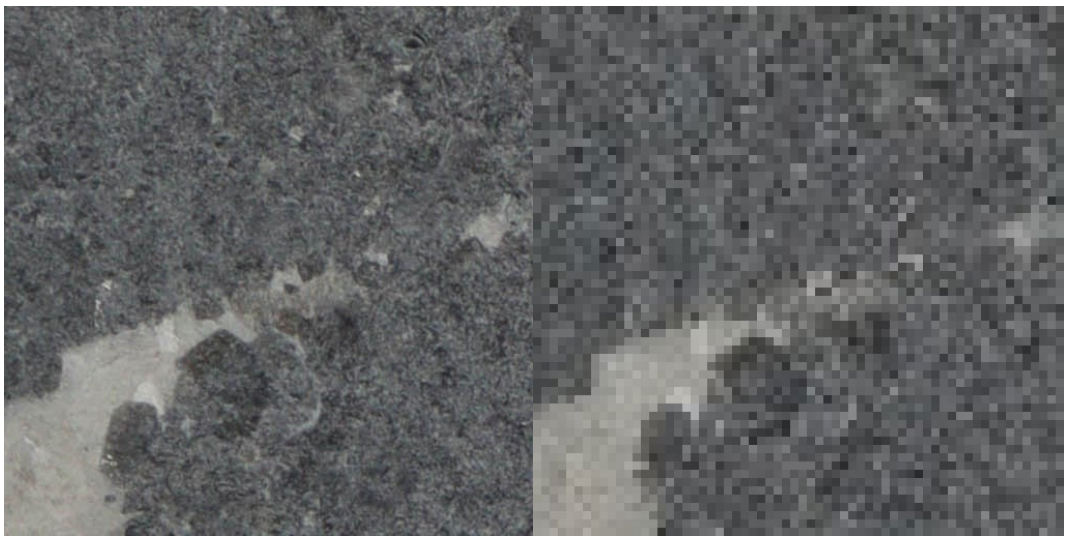
Figuur 8: Voorbeeld van de bepaalde contour en het uitgesneden oppervlak. De contour wordt weergeven door de rode rechthoek

2.4 Voorbewerkingen voor het convolutioneel netwerk

Het gebied van de steen dat gelokaliseerd is op de transportband wordt uitgesneden, zodanig dat de rand van de steen en de transportband wegvallen van de resterende afbeelding. Dit gebied wordt vervolgens opgedeeld in gelijke stukken ter grootte van 512 x 512 pixels. Deze stukken, van 512 x 512 pixels, worden vervolgens verkleind met behulp van dichtstbijzijnde interpolatie. Er werd ook getest met lineaire en oppervlakte interpolatie [10]. Bij gebruik van dichtstbijzijnde interpolatie komen kleine toevalligheden in de foto's meer naar boven. Aangezien we deze willen detecteren zorgt dit voor een beter resultaat in het convolutioneel netwerk.

2.4.1 Dichtstbijzijnde interpolatie bij het verkleinen van afbeeldingen

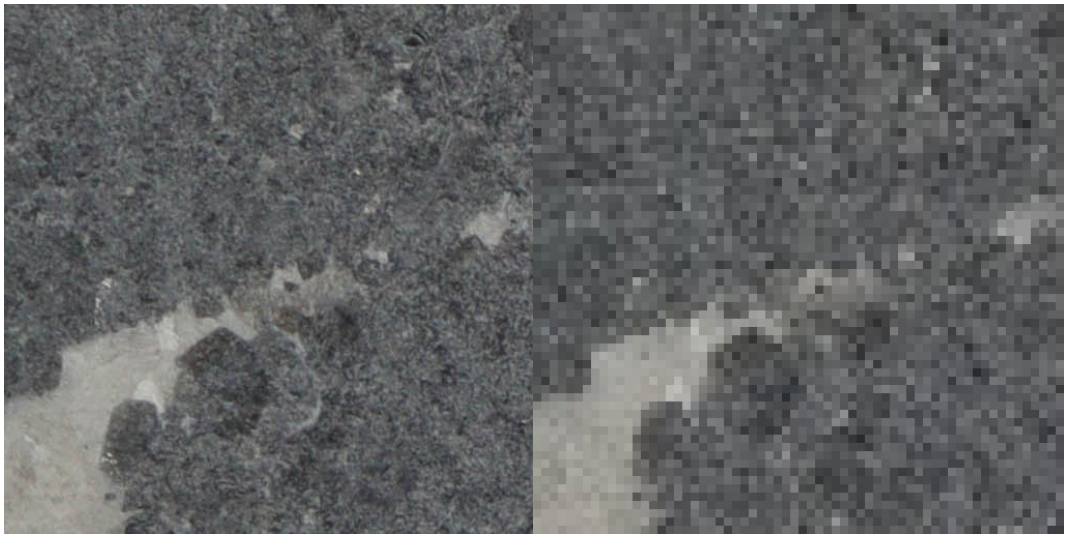
Dichtstbijzijnde interpolatie is een eenvoudige en snelle methode om afbeeldingen te verkleinen. Deze methode vervangt elke pixel in de originele afbeelding door de pixelwaarde van de meest nabijgelegen pixel in de verkleinde afbeelding [10], [11]. Figuur 9 geeft een voorbeeld van dichtstbijzijnde interpolatie.



Figuur 9: Voorbeeld van dichtstbijzijnde interpolatie. Deze afbeelding werd verkleind van 512x512 pixels naar 64x64 pixels

2.4.2 Lineaire interpolatie bij het verkleinen van afbeeldingen

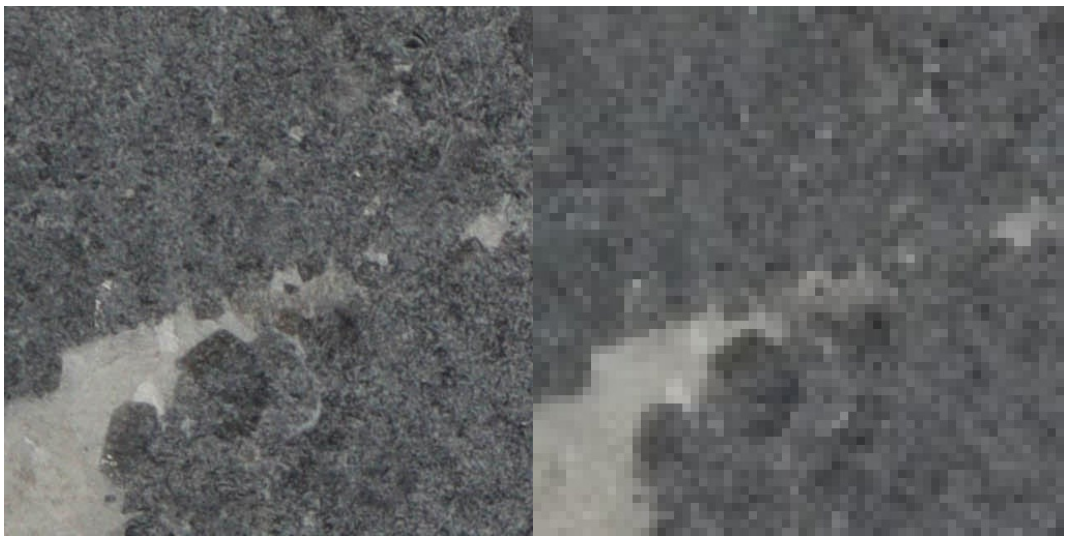
Lineaire interpolatie is een meer verfijnde techniek dan dichtstbijzijnde interpolatie voor het verkleinen van afbeeldingen. Bij lineaire interpolatie wordt de kleur van een nieuwe pixel bepaald door het gewogen gemiddelde te nemen van de pixelwaardes van de vier dichtstbijzijnde pixels in de originele afbeelding [10], [11]. Figuur 10 geeft een voorbeeld van lineaire interpolatie.



Figuur 10: Voorbeeld van lineaire interpolatie. Deze afbeelding werd verkleind van 512x512 pixels naar 64x64 pixels

2.4.3 Oppervlakte interpolatie bij het verkleinen van afbeeldingen

Oppervlakte interpolatie berekent de oppervlakte van de originele pixels en de oppervlakte van de nieuwe pixels. Vervolgens wordt de nieuwe pixelwaardes berekend door het gewogen gemiddelde van de originele pixelwaardes te nemen in functie van de overlapping met de nieuwe pixel [10], [11]. Figuur 11 geeft een voorbeeld van oppervlakte interpolatie.



Figuur 11: Voorbeeld van oppervlakte interpolatie. Deze afbeelding werd verkleind van 512x512 pixels naar 64x64 pixels

2.5 Training van het convolutioneel netwerk

2.5.1 Opstelling

Omwille van praktische redenen werd er voor het maken van de afbeeldingen die gebruikt werden voor het trainen van het convolutioneel netwerk een andere opstelling gebruikt dan voor de afbeeldingen van de stenen op de transportband. Figuur 12 geeft deze opstelling weer. Deze opstelling maakt ook gebruik van een spiegelreflexcamera van het type Canon EOS 250D gecombineerd met een Canon 18-55mm f/3.5-5.6 DC III lens. De camera-instellingen van de Canon EOS 250D in deze opstelling zijn als volgt:

- Diafragma: f/5.6.
- Belichtingstijd: 1/15 seconden
- ISO-waarde: 400
- Brandpuntafstand: 47 mm

Deze opstelling gebruikte hetzelfde LED-paneel als in de opstelling met de transportband.

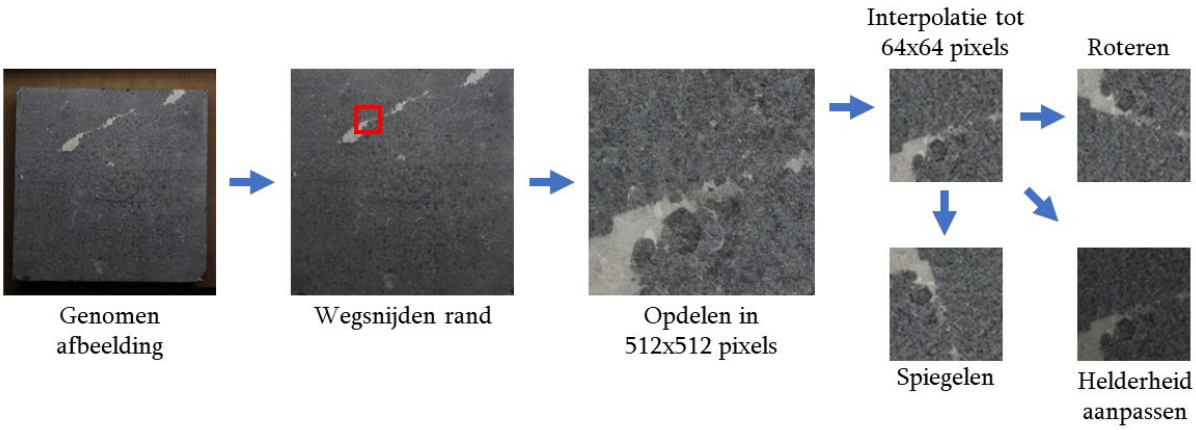
Figuur 12: Opstelling voor het maken van de afbeeldingen voor het convolutioneel netwerk

2.5.2 Voorbewerken van data

Als eerste stap wordt een vooraf gedefinieerde rand weggesneden om alleen het gebied van de steen over te houden. Dit maakt dat ongewenste achtergrondinformatie verwijderd wordt. Vervolgens wordt dit oppervlak opgedeeld in nieuwe afbeeldingen met een grootte van 512x512 pixels. Deze stap dient om de steen in beheersbare segmenten te verdelen voor verdere analyse en classificatie. Als derde stap worden de segmenten handmatig geclassificeerd in vijf mogelijke categorieën, namelijk: wit, zwart, fossiel, goed, moeilijk. Meer uitleg over deze categorieën is te vinden in 1.3 Doelstellingen. Deze segmenten worden als afbeeldingen opgeslagen in de map volgens de categorie. Vervolgens worden de segmenten verkleind, gebruikmakend van interpolatie, tot delen ter grootte van 64x64 pixels. Om ervoor te zorgen dat het convolutioneel netwerk minder gevoelig is aan toevalligheden worden de delen in alle mogelijke richtingen gespiegeld en geroteerd.

Vervolgens zal de helderheid van deze segmenten aangepast worden naar vijf verschillende niveaus. Deze verschillende helderheidsniveaus worden berekend door de originele pixelwaarden te vermenigvuldigen met een factor van 0,6 en 0,8 en 1 en 1,2 en 1,4. Het gebruiken van de verschillende helderheidsniveaus zorgt ervoor dat het convolutioneel netwerk minder gevoelig is aan veranderingen van het omgevingslicht.

Figuur 13 geeft de verschillende stappen weer op een schematische manier. Voor de eenvoud wordt er in Figuur 13 één aanpassing aan het helderheidsniveau weergeven.



Figuur 13: Schematisch overzicht van de voorbewerkingsstappen voor het convolutioneel netwerk

Al deze segmenten worden vervolgens verdeeld over een trainingsdataset en een validatiedataset. De trainingsdataset wordt gebruikt voor de effectieve training van het model. De validatie dataset wordt gebruikt om de performantie van het model te bepalen.

Zodra het convolutioneel netwerk een eerste keer berekend is, wordt dit netwerk gebruikt om alle geclassificeerde data te controleren op een foutieve classificatie. Wanneer een foutieve classificatie voorkomt, wordt deze gecorrigeerd door te herclassificeren.

2.5.3 Opbouw van het convolutioneel netwerk

Het convolutioneel netwerk maakt gebruik van verschillende lagen. Een eerste laag is de “rescaling” laag. Deze functie laat toe om de input van het convolutioneel netwerk die zich bevindt in het [0, 255] bereik om te zetten naar een [0, 1] bereik. Dit zorgt voor en standaardisatie in het netwerk [12].

Een volgende laag is de “Conv2D” laag. Deze laag is verantwoordelijk voor het uitvoeren van de convolutiebewerkingen op de ingevoerde segmenten. Deze functie maakt gebruik van een matrix die vermenigvuldigd wordt met de input van de laag. Deze matrix wordt een filter genoemd. Deze filter kan aangepast worden naargelang het gewenste resultaat van het netwerk [13].

Een derde laag is de “MaxPooling2D” laag. Deze laag laat toe om de dimensies te verkleinen van de tussenuitkomsten van het netwerk. Dit gebeurt met behoud van belangrijke kenmerken voor het model [14].

Een vierde laag is de “Flatten” laag. Deze laag zet de multidimensionale input laag om naar een eendimensionale vector [15].

Een laatste laag is de “Dense” laag. Deze laag is een volledig geconnecteerde laag. Dit betekent dat de hele input van deze laag volledig verbonden is met de uitgangen van deze laag gebruikmakend van een lineaire wegingsfactor. Deze laat dus toe om over de gehele input lineaire berekeningen te doen [16].

2.6 Gebruik van het convolutioneel netwerk

Het getrainde convolutioneel netwerk kan opgeslagen worden met behulp van de functie “model.save()” van de bibliotheek Keras, onderdeel van tensorflow. Vervolgens kan het convolutioneel netwerk ingeladen worden met de functie “models.load()” van de bibliotheek Keras. Aangezien het oppervlak van de steen opgedeeld wordt in segmenten met een raster van 512x512 pixels zullen deze segmenten verkleind moeten worden met dezelfde interpolatie methode als gebruikt is voor het trainen van het convolutioneel netwerk. Deze bewerking zorgt dat de inputafbeelding bestaat uit een rasteren van 64 x64 pixels. De output van het convolutioneel netwerk bestaat uit een lijst van vijf elementen. Deze elementen zijn de waarschijnlijkheden die overeenkomen met de vijf verschillende mogelijke uitkomsten van het convolutioneel netwerk. Op basis van de waarschijnlijkheden wordt vervolgens de overeenkomstige klasse bepaald.

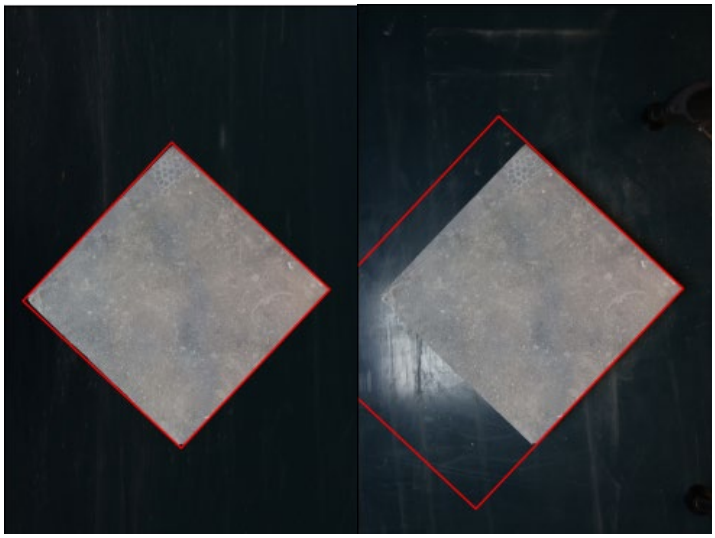
2.7 Afgewerkt programma

Het afgewerkt programma maakt gebruik van de tkinter bibliotheek. Deze bibliotheek laat toe op een eenvoudige manier zowel de lokalisatie als ook de uitkomsten van het convolutioneel netwerk te visualiseren. Deze visualisatie dient ook de score per categorie weer te geven alsook de rekentijd van het convolutioneel netwerk.

3 Resultaten

3.1 Lokalisatie

Figuur 14 geeft een resultaat van het voorgestelde lokalisatiemethode weer. De linker figuur heeft geen last van reflecties op de transportband, de rechter figuur heeft hier wel last van. De rode vierhoeken geven weer waar het programma denkt een steen te lokaliseren. De reflectie op de transportband zorgt dus voor een foute lokalisatie. Dit zorgt ook voor fouten in het tweede gedeelte van het programma waar de defecten gelokaliseerd moeten worden. Het is dus van groot belang deze reflecties te voorkomen. Er werd getest met verschillende belichtingen. Het LED-paneel bleek de beste gelijke belichting te voorzien waardoor er minder reflecties aanwezig waren.



Figuur 14: Lokalisatie van dezelfde steen zonder (links) en met (rechts) reflecties van de transportband

3.2 Training van het convolutioneel netwerk

Voor de training van het convolutioneel netwerk maakte deze thesis gebruik van de drie interpolatietechnieken, die besproken zijn in 2.4 Voorbewerkingen voor het convolutioneel netwerk. Deze interpolatietechnieken worden vergeleken op basis van trainingsnauwkeurigheid, trainingsverliezen, validatienauwkeurigheid en validatieverliezen.

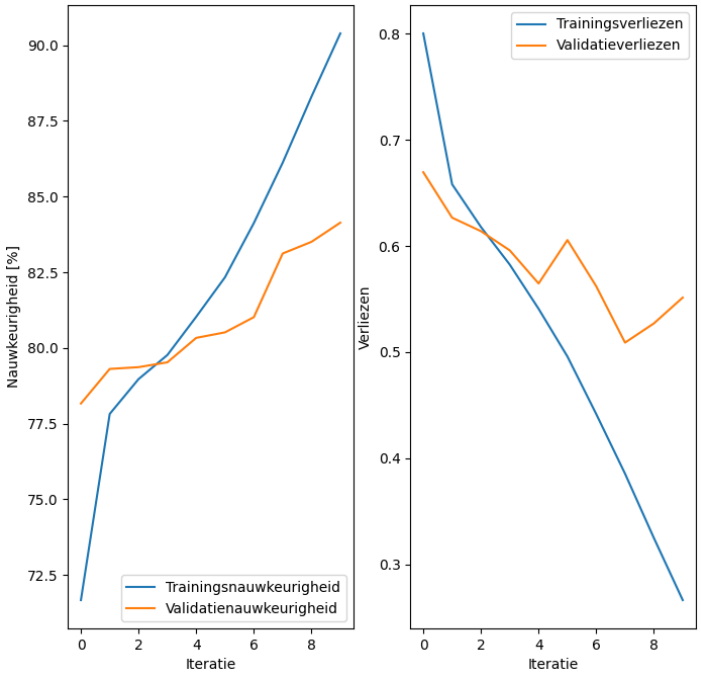
Vervolgens wordt er ook vergelijken indien helderheidsaanpassingen een positieve invloed hebben op het convolutioneel netwerk. Als laatste stap werd er ook een herclassificatie toegepast van de gemaakte data op basis van het origineel best werkende convolutioneel netwerk.

Voor de training van het netwerk werd er gebruik gemaakt van een server met een Intel(R) Xeon(R) CPU E5-2680 v2 processor. Het trainen van het convolutioneel netwerk zonder het gebruik van verschillende helderheden nam een tijd van 432 seconden in beslag. Het trainen van het convolutioneel netwerk met verschillende helderheden nam een tijd van 2158 seconden in beslag.

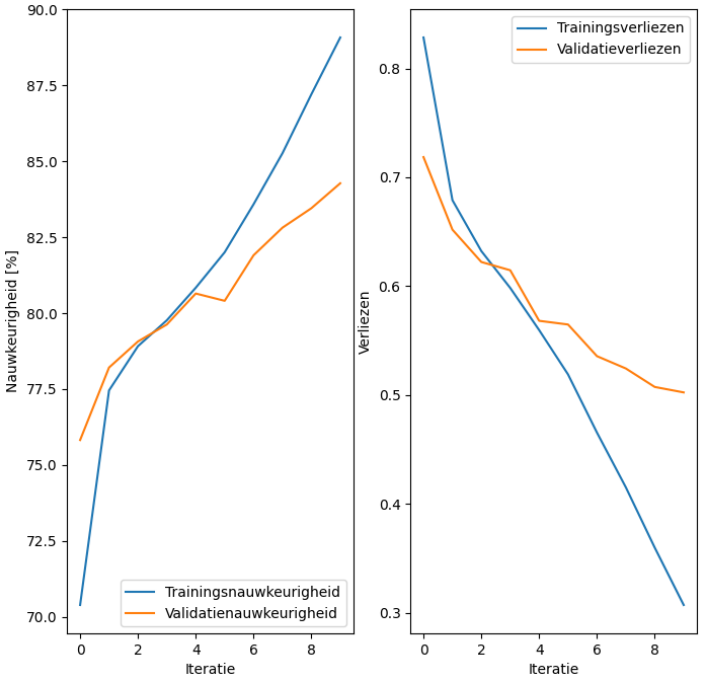
3.2.1 Interpolatietechnieken

Als eerste werd er getest met de verschillende interpolatietechnieken. De interpolatietechnieken met de uitleg omtrent de werking zijn te vinden in 2.4 Voorbewerkingen voor het convolutioneel netwerk. Figuur 15 geeft de nauwkeurigheid en verliezen weer bij het trainen en valideren van het

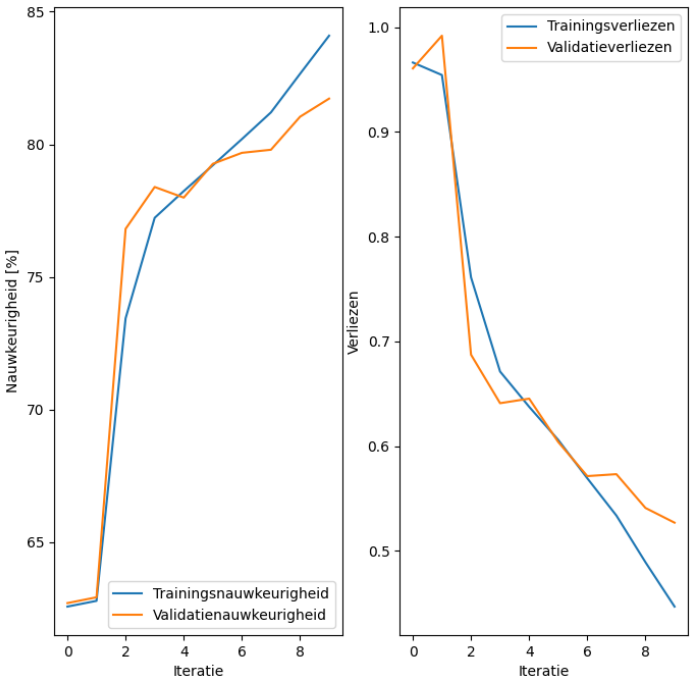
convolutioneel netwerk op basis van dichtstbijzijnde interpolatie. Figuur 16 geeft de nauwkeurigheid en verliezen weer bij het trainen en valideren van het convolutioneel netwerk op basis van lineaire interpolatie. Figuur 17 geeft de nauwkeurigheid en verliezen weer bij het trainen en valideren van het convolutioneel netwerk op basis van oppervlakte interpolatie. Tabel 1 geeft de resultaten van de nauwkeurigheid en verliezen weer bij het trainen en valideren van de geteste interpolatietechnieken bij de tiende iteratie.



Figuur 15: Nauwkeurigheid en verliezen ten op zichten van het aantal iteratie bij het trainen en valideren van het convolutioneel netwerk op basis van dichtstbijzijnde interpolatie



Figuur 16: Nauwkeurigheid en verliezen ten op zichten van het aantal iteratie bij het trainen en valideren van het convolutioneel netwerk op basis van lineaire interpolatie



Figuur 17: Nauwkeurigheid en verliezen ten op zichten van het aantal iteratie bij het trainen en valideren van het convolutioneel netwerk op basis van oppervlakte interpolatie

Tabel 1: Resultaten van training van het convolutioneel netwerk gebaseerd op de drie verschillende interpolatietechnieken na tien iteraties

	Dichtstbijzijnde interpolatie	Lineaire interpolatie	Oppervlakte interpolatie
Trainingsnauwkeurigheid [%]	90,39	89,07	84,09
Validatienauwkeurigheid [%]	84,27	84,13	81,72
Trainingsverliezen	0,26	0,31	0,45
Validatieverliezen	0,5	0,55	0,53

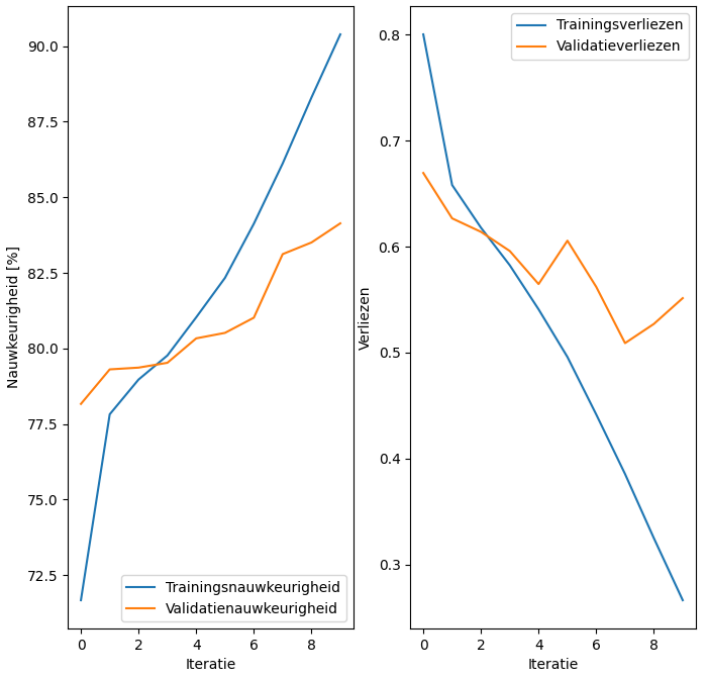
Bij het vergelijken van de drie bovenstaande figuren is zichtbaar hoe de drie types iteraties ongeveer met gelijke nauwkeurigheid starten. De snelheid waarmee de nauwkeurigheid verbeterd is sneller bij dichtstbijzijnde en lineaire interpolatie. Uit de eindresultaten in Tabel 1 blijkt dat zowel de trainings- en validatienauwkeurigheden bij Dichtstbijzijnde en Lineaire interpolatie vrij dicht bij elkaar liggen, en beide beter presteren na 10 iteraties dan Oppervlakte interpolatie.

De verliezen zijn getallen die aangeven hoe ver de predictie afzit van de werkelijke waarde. Indien deze nul zouden zijn wil dit zeggen dat er altijd perfecte predicties gemaakt worden. Het is dus gewenst dat deze getalen zo dicht mogelijk bij nul liggen.

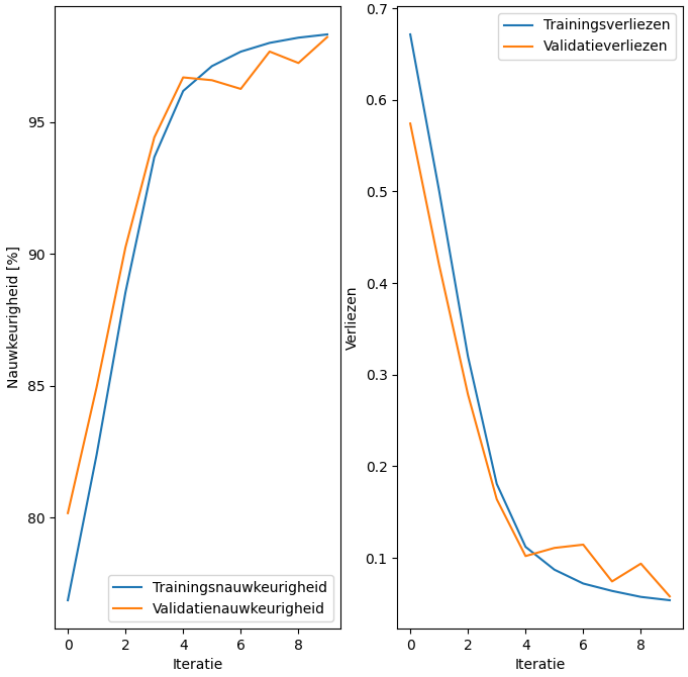
Aangezien de trainingsnauwkeurigheid en trainingsverliezen het beste zijn bij dichtstbijzijnde interpolatie, wordt er gekozen om deze techniek te gebruiken doorheen de rest van het onderzoek. De validatieresultaten zijn gelijkaardig bij de 3 interpolaties.

3.2.2 Helderheidsaanpassingen

Als tweede werd er getest indien helderheidsaanpassingen een positieve invloed hebben op het convolutioneel netwerk. Figuur 18 geeft de nauwkeurigheid en verliezen weer bij het trainen en valideren van het convolutioneel netwerk op basis van dichtstbijzijnde interpolatie zonder helderheidsaanpassingen. Figuur 19 de nauwkeurigheid en verliezen weer bij het trainen en valideren van het convolutioneel netwerk op basis van dichtstbijzijnde interpolatie met helderheidsaanpassingen. Tabel 2 geeft de resultaten van de nauwkeurigheid en verliezen weer bij het trainen en valideren van het al dan niet gebruiken van helderheidsaanpassingen bij de tiende iteratie.



Figuur 18: Nauwkeurigheid en verliezen ten op zichten van het aantal iteratie bij het trainen en valideren van het convolutioneel netwerk op basis van dichtstbijzijnde interpolatie zonder helderheidsaanpassingen



Figuur 19: Nauwkeurigheid en verliezen ten op zichten van het aantal iteratie bij het trainen en valideren van het convolutioneel netwerk op basis van dichtstbijzijnde interpolatie met helderheid aanpassingen

Tabel 2: Resultaten van training van het convolutioneel netwerk gebaseerd op geen helderheid aanpassen en training van het convolutioneel netwerk met vijf helderheidaanpassingen

	Zonder helderheidsaanpassingen	Met helderheidsaanpassingen
Trainingsnauwkeurigheid [%]	90,39	98,56
Validatienauwkeurigheid [%]	84,27	97,2
Trainingsverliezen	0,26	0,05
Validatieverliezen	0,5	0,1

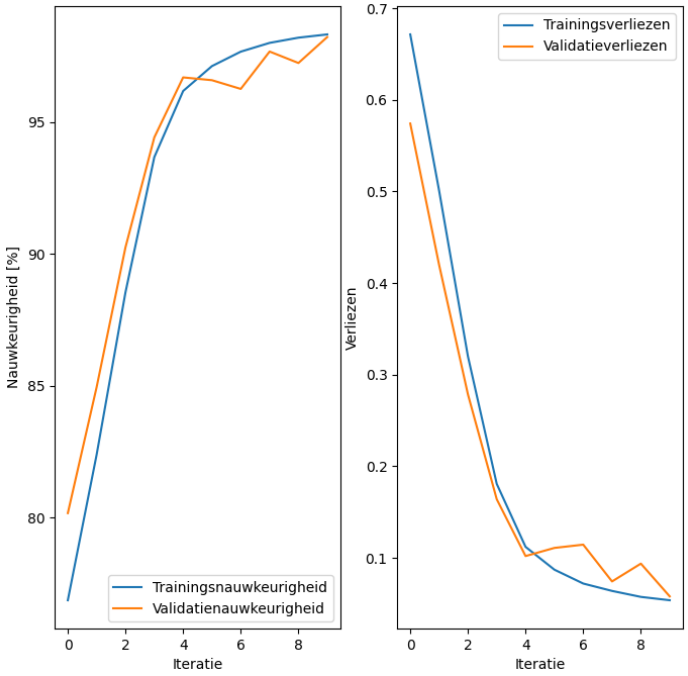
Een convolutioneel netwerk maakt voorspellingen afhankelijk van de trainingsdata die deze gekregen heeft. Des te meer verschillende data het netwerk krijgt, des te beter het dus gaat presteren. Het aanpassen van de helderheid wordt in 5 stappen gedaan. Bij elke stap wordt er een kopie gemaakt van de afbeelding. De originele helderheid wordt veranderd met een factor (0.6), (0.8), (1), (1.2) en (1.4). Doordat het aantal beelden dus met een factor vijf vermeerderd door de helderheid aan te passen heeft het netwerk ook vijf keer zoveel data om mee te trainen.

De stenen die gecontroleerd moeten worden op defecten worden uit de natuur gehaald. Niet alle stenen in de natuur hebben dezelfde tint grijs. De helderheidsaanpassing geeft het convolutioneel netwerk dus ook meer data van andere kleuren grijs. Het convolutioneel netwerk is hierdoor beter in staat verschillende kleuren stenen toch in de juiste klasse in te delen.

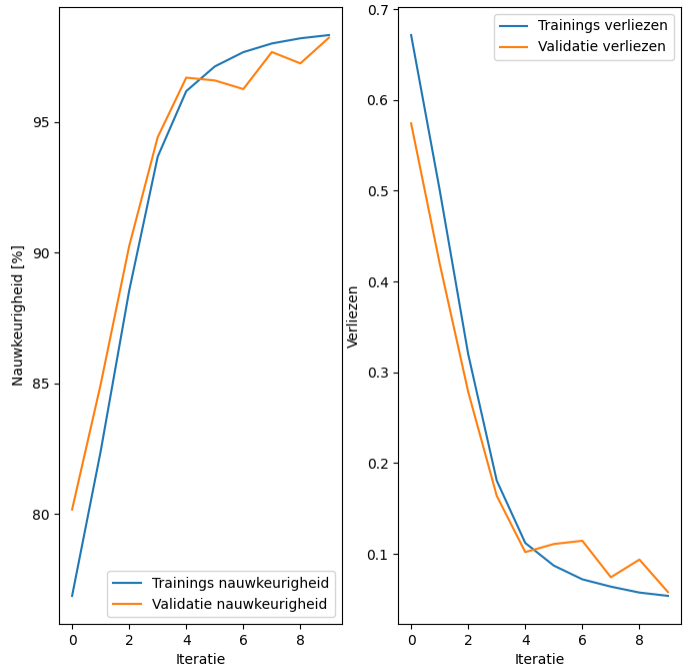
Zoals het vorige puntje vermeldt, is er bij het werken met een convolutioneel netwerk gewenst zo laag mogelijke verliezen te hebben en een zo hoog mogelijke nauwkeurigheid. Tabel 2 geeft een duidelijk verschil weer tussen beide gevallen, zonder of met toevoeging van de extra beelden door helderheidsaanpassingen. De nauwkeurigheid stijgt bij zowel de trainingsset als validatieset met 10 tot 15 procent. De verliezen dalen eveneens met een factor 5. Uit deze resultaten kan besloten worden dat het toevoegen van de extra data een meerwaarde is voor het programma.

3.2.3 Herclassificatie

Als laatste werd er getest of het handmatig herclassificeren van de stenen een positieve invloed heeft op het convolutioneel netwerk. Figuur 20 geeft de nauwkeurigheid en verliezen weer bij het trainen en valideren van het convolutioneel netwerk op basis van dichtstbijzijnde interpolatie met helderheidsaanpassingen zonder herclassificatie. Figuur 21 geeft de nauwkeurigheid en verliezen weer bij het trainen en valideren van het convolutioneel netwerk op basis van dichtstbijzijnde interpolatie met helderheidsaanpassingen met herclassificatie. Deze herclassificatie is gebaseerd op het convolutioneel netwerk gebruik maken van dichtstbijzijnde interpolatie zonder helderheidsaanpassingen. Tabel 3 geeft de resultaten van de nauwkeurigheid en verliezen weer bij het trainen en valideren van het convolutioneel netwerk al dan niet met herclassificatie bij de tiende iteratie.



Figuur 20: Nauwkeurigheid en verliezen ten op zichten van het aantal iteratie bij het trainen en valideren van het convolutioneel netwerk op basis van dichtstbijzijnde interpolatie met helderheid aanpassingen zonder herclassificatie



Figuur 21: Nauwkeurigheid en verliezen ten op zichten van het aantal iteratie bij het trainen en valideren van het convolutioneel netwerk op basis van dichtstbijzijnde interpolatie met helderheid aanpassingen met herclassificatie

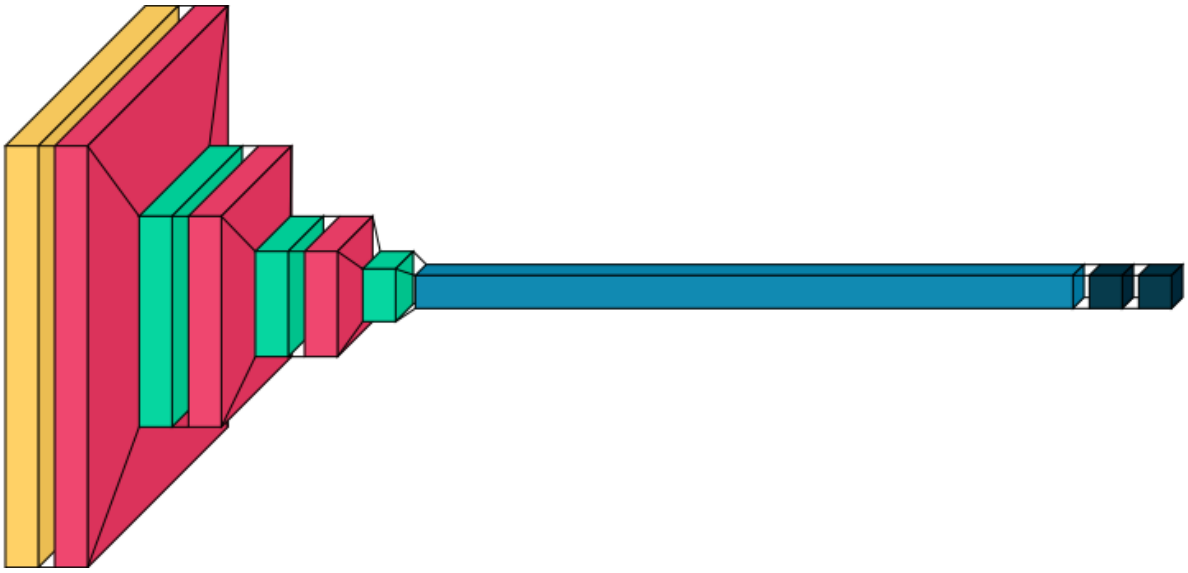
Tabel 3: Resultaten van training van het convolutioneel netwerk gebaseerd met of zonder herclassificatie

	Zonder herclassificatie	Met herclassificatie
Trainingsnauwkeurigheid [%]	98,56	98,31
Validatienauwkeurigheid [%]	97,2	98,2
Trainingsverliezen	0,05	0,05
Validatieverliezen	0,1	0,06

Tabel 3 geeft weer dat er een heel kleine stijging is van de validatienauwkeurigheid en een kleine daling van de validatieverliezen. De trainingsresultaten blijven nagenoeg gelijk. Het herclassificeren heeft dus gezorgd voor een lichte verbetering van de prestatie van het convolutioneel netwerk. De mogelijkheid bestaat dat bepaalde stenen die in eerste instantie juist geclassificeerd werden nu niet meer in de juiste klasse worden ingedeeld na de herclassificatie. Om nog enkel kleine verbeteringen te maken aan het programma zou deze herclassificatiestap nog enkel keren uitgevoerd kunnen worden. Dit is een zeer tijdrovend proces dat voor deze thesis een minimale meerwaarde zou zijn.

3.2.4 Opbouw netwerk

Figuur 22 geeft een schematisch overzicht van het getrainde convolutioneel netwerk. De verschillende typen lagen zijn te vinden in 2.5.3 Opbouw van het convolutioneel netwerk.



Figuur 22: Schematisch overzicht van het getrainde convolutioneel netwerk waarbij een gele laag een "Rescaling"-laag voorstelt, een roze laag overeenkomt met een "Conv2D"-laag, een groene laag overeenkomt met een "MaxPooling2D"-laag, de lichtblauwe laag overeenkomt met een "Flatten"-laag en de donkerblauwe laag overeenkomt met een "Dense"-laag

Tabel 4: De verschillende lagen in het convolutioneel netwerk

	Laag type
Laag 1	Rescaling
Laag 2	Conv2D met filtermatrix van 16x16
Laag 3	MaxPooling2D
Laag 4	Conv2D met filtermatrix van 32x32
Laag 5	MaxPooling2D
Laag 6	Conv2D met filtermatrix van 64x64
Laag 7	MaxPooling2D
Laag 8	Flatten
Laag 9	Dense met 128 eenheden
Laag 10	Dense met 5 eenheden

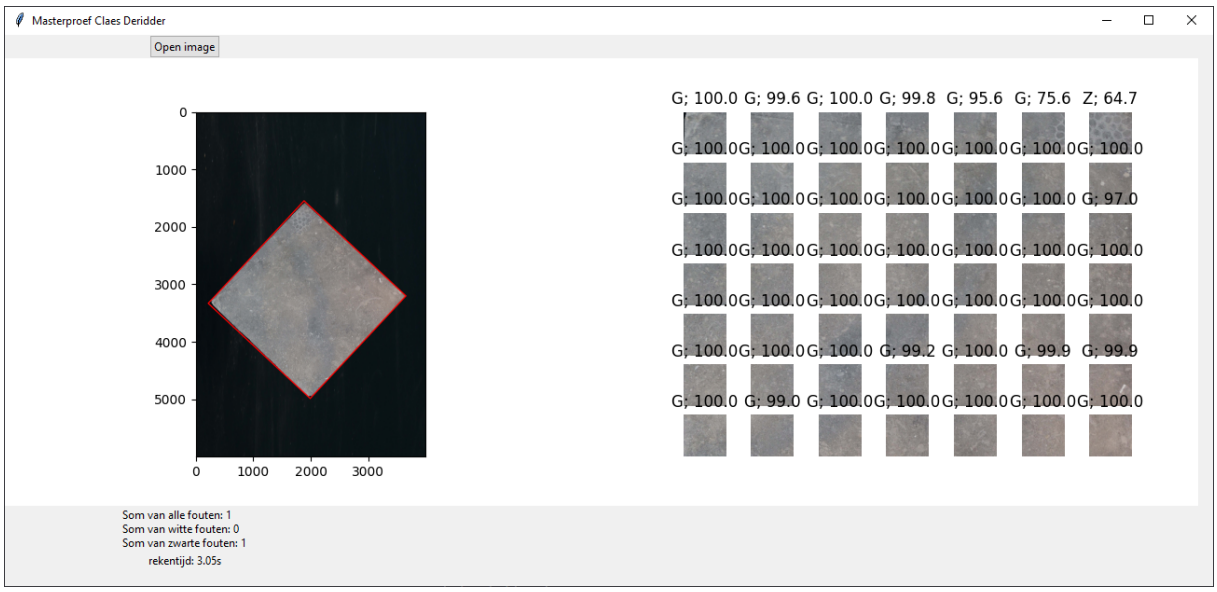
3.3 Eindprogramma

Figuur 23 geeft een voorstelling van het eindprogramma weer. Dit eindprogramma is opgebouwd met behulp van de bibliotheek tkinter. De input van het eindprogramma is een afbeelding van een steen op de transportband. Wanneer deze afbeelding ingeladen wordt maakt het eindprogramma gebruik van de lokalisatiemethode, zoals reeds beschreven in 3.1 Lokalisatie. Deze lokalisatie is weergeven links door Figuur 23.

Vervolgens splits het eindprogramma het gelokaliseerde steenoppervlak op in segmenten van 512x512 pixels. Deze segmenten worden vervolgens met de gekozen “dichtstbijzijnde" interpolatiemethode verkleind tot 64x64 pixels.

De verkleinde segmenten worden gebruikt als input voor het convolutioneel netwerk. Dit netwerk zal vervolgens een voorspelling maken van welk defect aanwezig is in dit deelsegment. Elk defect heeft een specifieke categorie die wordt weergegeven met een bijhorende letter. G duidt een voorspelling “Goed” aan, Z betekend een voorspelling “Zwart”, W een voorspelling “Wit”, F een voorspelling “Fossiel” en tot slot wordt M toegekend bij een voorspelling “Moeilijk”. Het getal dat bij de toegekende categorie staat is de zekerheid die het convolutioneel netwerk heeft over de gegeven voorspelling. Aan de rechterkant van Figuur 23 is zichtbaar hoe de steen is opgedeeld in alle deelsegmenten. Boven elk segment wordt de bijhorende voorspelling en zekerheid weergegeven.

Linksonder op Figuur 23 zijn de sommen per fout weergeven. Deze worden berekend door het aantal fouten te vermenigvuldigen met de respectievelijke wegingsfactor per fout. In het voorbeeld in Figuur 22 staan de factoren ingesteld op 10 voor de voorspelling “Fossielen”, 1 voor een voorspelling “Wit”, 1 voor een voorspelling “Zwart” en 10 voor de voorspelling “Moeilijk”.



Figuur 23: Het afgewerkte programma waarbij de linker afbeelding de lokalisatie weergeeft en de rechter afbeelding de uitkomsten van het convolutioneel netwerk

4 Besluit

Deze thesis stelt een lokalisatie methode voor die, gebruikmakend van een threshold, de contour van een steen op een transportband kan bepalen op basis van de grijswaardenafbeelding van deze steen. Deze contour wordt vervolgens gebruikt voor het roteren van de afbeelding zodat de contour zo parallel mogelijk georiënteerd ligt aan de grenzen van de afbeelding. Vervolgens kan deze steen op basis van de contour uitgesneden worden.

In de volgende stap wordt dit beeld verder opgedeeld in segmenten van 512x512 pixels. Deze segmenten dienen de grootte te hebben van het convolutioneel netwerk. Het convolutioneel netwerk verwacht een segment met een grootte van 64x64 pixels. Om dit te realiseren werd er in deze thesis getest met drie interpolatie technieken, namelijk: dichtstbijzijnde interpolatie, lineaire interpolatie en oppervlakte interpolatie. Deze interpolatiemethodes werden vergeleken op basis van de validatienauwkeurigheid. De validatienauwkeurigheden waren, respectievelijk, 84,27%, 84,13% en 81,72%. Dichtstbijzijnde interpolatie toont de hoogste nauwkeurigheid en werd daarom gebruikt als interpolatiemethode. De meest significante verbetering was het toevoegen van de verschillende helderheidsklassen. Deze toevoeging liet de validatienauwkeurigheid stijgen tot 97,2%.

Vervolgens werd de lokalisatie en het getrainde convolutioneel netwerk samengebracht tot één geheel, gebruikmakend van de bibliotheek tkinter. Dit eindprogramma laat toe om de lokalisatie van de steen, de opsplitsing tot segmenten van 512x512 pixels, de uitkomst van het convolutioneel netwerk per segment en de gewogen som te visualiseren.

Naar de toekomst toe is er de mogelijkheid om het geheel te verbeteren. Voor de lokalisatie kunnen er andere technieken gebruikt worden die de contour van de steen met een hogere nauwkeurigheid kunnen bepalen. Een voorbeeld hiervan is te gebruik maken van vormdetectie. Bij een goede implementatie kan dit resulteren in het beter detecteren van de stenen wanneer er een aanpassing in het omgevingslicht of wanneer er zich reflecties voordoen op de transportband.

Vervolgens kan de nauwkeurigheid van het convolutioneel netwerk verbeterd worden door het toevoegen van extra data tijdens het trainen van het netwerk. Deze extra data kunnen bestaan uit varianten van de reeds gekende data of door het toevoegen van volledig nieuwe data. De voorkeur hier gaat uit naar het toevoegen van volledig nieuwe data zodat het convolutioneel netwerk dan meer getraind is op variaties binnen de verschillende categorieën.

De snelheid van het uiteindelijke programma kan ook verbeterd worden. Het gebruik van meerdere threads, tijdens de berekening door het convolutioneel netwerk, kan het programma aanzienlijk versnellen. Er kan ook geopteerd worden om gebruik te maken van een andere programmeertaal, bijvoorbeeld: C++ of C#, aangezien dat deze programmeertalen sneller kunnen berekenen.

Referentielijst

[1] "Over ons | Optidrive." https://www.optidrive.be/nl/over-ons (accessed Nov. 25, 2022).

[2] M. Brejl and M. Sonka, "Object localization and border detection criteria design in edge-based image segmentation: Automated learning from examples," *IEEE Trans Med Imaging*, vol. 19, no. 10, pp. 973–985, Oct. 2000, doi: 10.1109/42.887613.

[3] M. E. Celebi *et al.*, "Border detection in dermoscopy images using statistical region merging," *Skin Research and Technology*, vol. 14, no. 3, pp. 347–353, Aug. 2008, doi: 10.1111/j.1600-0846.2008.00301.x.

[4] P. Lassalle, J. Inglada, J. Michel, M. Grizonnet, and J. Malik, "A Scalable Tile-Based Framework for Region-Merging Segmentation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 10, pp. 5473–5485, Oct. 2015, doi: 10.1109/TGRS.2015.2422848.

[5] J. H. Xue and D. M. Titterington, "T-Tests, F-Tests and Otsu's methods for image thresholding," *IEEE Transactions on Image Processing*, vol. 20, no. 8, pp. 2392–2396, Aug. 2011, doi: 10.1109/TIP.2011.2114358.

[6] J. Canny, "A Computational Approach to Edge Detection," *IEEE Trans Pattern Anal Mach Intell*, vol. PAMI-8, no. 6, pp. 679–698, 1986, doi: 10.1109/TPAMI.1986.4767851.

[7] "What are Neural Networks? | IBM." https://www.ibm.com/cloud/learn/neural-networks (accessed Nov. 25, 2022).

[8] "Welcome to Python.org." https://www.python.org/ (accessed May 04, 2023).

[9] "Project Jupyter | Home." https://jupyter.org/ (accessed May 04, 2023).

[10] "OpenCV: Geometric Image Transformations." https://docs.opencv.org/4.7.0/da/d54/group__imgproc__transform.html (accessed May 08, 2023).

[11] D. Han, "Comparison of Commonly Used Image Interpolation Methods," pp. 1556–1559, Mar. 2013, doi: 10.2991/ICCSEE.2013.391.

[12] "Rescaling layer." https://keras.io/api/layers/preprocessing_layers/image_preprocessing/rescaling/ (accessed May 24, 2023).

[13] "Conv2D layer." https://keras.io/api/layers/convolution_layers/convolution2d/ (accessed May 24, 2023).

[14] "MaxPooling2D layer." https://keras.io/api/layers/pooling_layers/max_pooling2d/ (accessed May 24, 2023).

[15] "Flatten layer." https://keras.io/api/layers/reshaping_layers/flatten/ (accessed May 24, 2023).

[16] "Dense layer." https://keras.io/api/layers/core_layers/dense/ (accessed May 24, 2023).

Bijlagen

Geïnstalleerde bibliotheken in de python omgeving

Bibliotheek	Versie
absl-py	1.4.0
astunparse	1.6.3
cachetools	5.3.0
certifi	2022.12.7
charset-normalizer	3.1.0
contourpy	1.0.7
cycler	0.11.0
flatbuffers	23.3.3
fonttools	4.39.3
gast	0.4.0
google-auth	2.17.3
google-auth-oauthlib	1.0.0
google-pasta	0.2.0
grpcio	1.54.0
h5py	3.8.0
idna	3.4
jax	0.4.8
keras	2.12.0
kiwisolver	1.4.4
libclang	16.0.0
Markdown	3.4.3
MarkupSafe	2.1.2
matplotlib	3.7.1
ml-dtypes	0.1.0
numpy	1.23.5
oauthlib	3.2.2
opencv-python	4.7.0.72
opt-einsum	3.3.0
packaging	23.1

Bibliotheek	Versie
Pillow	9.5.0
pip	22.2.2
protobuf	4.22.3
pyasn1	0.5.0
pyasn1-modules	0.3.0
pyparsing	3.0.9
python-dateutil	2.8.2
requests	2.28.2
requests-oauthlib	1.3.1
rsa	4.9
scipy	1.10.1
setuptools	63.2.0
six	1.16.0
tensorboard	2.12.2
tensorboard-data-server	0.7.0
tensorboard-plugin-wit	1.8.1
tensorflow	2.12.0
tensorflow-estimator	2.12.0
tensorflow-intel	2.12.0
tensorflow-io-gcs-filesystem	0.31.0
termcolor	2.3.0
thread6	0.2.0
tk	0.1.0
typing_extensions	4.5.0
urllib3	1.26.15
Werkzeug	2.3.0
wheel	0.40.0
wrapt	1.14.1