

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

Michiel Claes

Deridder Dries

master IW Elektromechanica

master IW Elektromechanica

Situering

Samenwerking **Optidrive** Leuven, specialisatie: ontwerpen en realiseren van geautomatiseerde robotinstallaties voor natuursteenverwerkende bedrijven.

- Classificatie al mogelijk op basis van steengrootte
- Vraag naar systeem voor automatische kwaliteitscontrole
- Moeilijkheid: variatie van stenen in kleur, dikte, kwaliteit en afwerking

DOEL: Ontwikkeling software programma voor steencontrole en onderverdeling in kwaliteitsklassen.

2 vereisten ontwikkelen programma

lokaliseren natuursteen op transportband door camera en sensoren

Detecteren van defecten in stenen door analyseren steenoppervlakte

Materiaal en methode



Programmeertaal: Python

Opstelling: machine met 2 transportbanden beschikbaar waarrond een tijdelijke opstelling werd gebouwd.

Verzamelen beeldmateriaal:

- Stenen op transportband voor **Lokalisatie**
- Afzonderlijke oppervlakken voor **Detectie van fouten** → zodat het convolutioneel netwerk genoeg data heeft om zichzelf te trainen.

Figuur 1 : Opstelling maken van beelden

Lokalisatie

Bij het opbouwen van de foto-opstelling werd rekening gehouden met een duidelijk kleurcontrast tussen de stenen en het oppervlak van de transportband.

Stap 1: beelden omzetten in grijswaarden.

Stap 2: de waarde van elke afzonderlijke pixel wordt vergeleken met een threshold, deze bepaalt of de pixel zwart of wit gekleurd wordt.

Stap 3: Met behulp van randdetectie wordt de grootste contour geselecteerd. Deze selectie geeft de gezochte steen weer.

Stap 4: Vanaf dat de steen gelokaliseerd is wordt deze geroteerd totdat de randen parallel staan.

Stap 5: Uitknippen van steenoppervlak.



Figuur 2 : stap 3 t.e.m. 5

Detectie van fouten

Trainen Programma:

De stenen kunnen op verschillende plaatsen verschillende defecten hebben. Om detectie te vergemakkelijken wordt het steenoppervlak opgesplitst in kleinere deelsegmenten → zorgt voor meer trainingsdata → vermindert de nodige grootte van het neurale netwerk.

Deze afbeeldingen werden handmatig geclassificeerd:



Figuur 3 : Klasse Fossiel



Figuur 4 : Klasse Zwart



Figuur 5 : Klasse Wit



Figuur 6 : Klasse Goed

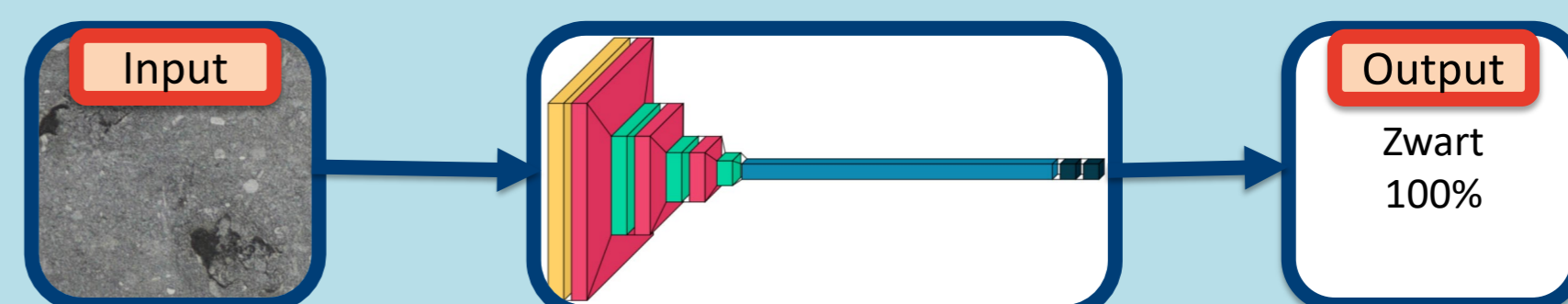
Om extra beelden te verkrijgen werden de beelden in alle mogelijke combinaties gespiegeld en 90° geroteerd. Ook de helderheid van de afbeeldingen werd als variabele aangepast.

Uitvoeren Programma:

Met behulp van lineaire-, dichtstbijzijnde- en oppervlakte-interpolatie worden de initiële beelden uiteindelijk verkleind tot 64 op 64 pixels. Dit is de input voor het convolutioneel neurale netwerk. Het aantal layers en nodes van het netwerk werd proefondervindelijk aangepast.

De steen wordt opgedeeld in deelsegmenten. Elk segment wordt gescand op zoek naar defecten. De output van het netwerk is een opgedeelde afbeelding, die weergeeft in welk segment welk defect aanwezig is met bijhorend zekerheidspercentage.

Met behulp van een wegingsfactor, verschillend per defect, krijgt elk segment een score. De totaalscore van de steen is een weergave van de kwaliteit van de steen.



Figuur 7 : Schema van neurale netwerk

Resultaten en Conclusie

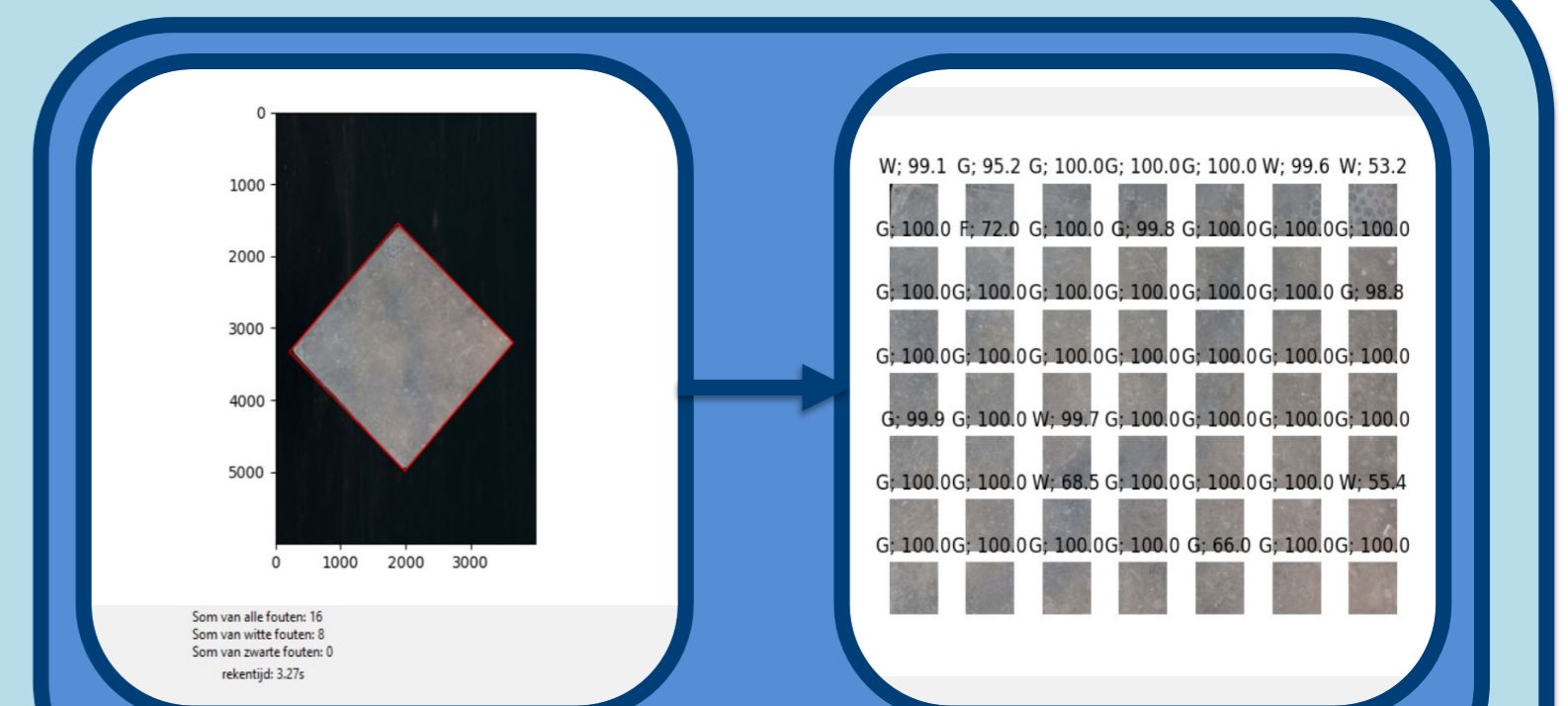
In deze thesis was het doel een programma te ontwikkelen dat natuurstenen kon opdelen in verschillende kwaliteitsklassen. De 2 vereiste onderdelen van het programma werden afzonderlijk geschreven en vervolgens samengevoegd tot één programma.

Conclusie:

Uit de verschillende interpolaties bleek dichtstbijzijnde interpolatie de beste resultaten te geven, bij deze techniek komen kleine toevalligheden in de foto's meer naar boven.

Aangezien dit de fouten zijn die gedetecteerd moeten worden zorgt dit voor een beter resultaat in het convolutioneel netwerk. Op het rechtse deel van figuur 8 is zichtbaar hoe elk segment van de steen afzonderlijk een klasse toegekend krijgt. De som van de weegfactoren per klasse worden onderaan de figuur weergegeven.

Het getrainde convolutioneel netwerk geeft volgende resultaten: **accuracy: 98,31 / loss: 0,06**



Figuur 8 : in- en output python programma

Promotoren / Copromotoren / Begeleiders

Prof. dr. ir. Johan Baeten
Dhr. Jan Van Elsacker