



VEILIGE EN EFFICIËNTE INSPECTIE VAN HET SPOOR

Met het project 'Veilige en efficiënte inspectie van het spoor' wonnen de bedrijven Inspection' en en CQM² de Hendrik Lorentz Prijs³. In dit project wordt onder meer deep-learningtechniek intelligent ingezet in combinatie met visualisatietechnieken waardoor spoorweginspecteurs veel sneller en effectiever gebreken in het spoor kunnen identificeren en controleren.*

JOHAN VAN ROOIJ & HUUB VAN DEN BROEK

Veilige semiautomatische inspectie van het spoor

Nederland heeft het drukst bereden spoornet van Europa. Gemiddeld maken reizigers elke dag 1,1 miljoen treinreizen – in totaal 17 miljard reizigerskilometers per jaar. Het is dan ook van groot belang zorgvuldig en efficiënt met het spoor om te gaan. Door monitoring en inspectie van spoorstaven en wissels is het mogelijk beginnende defecten aan het spoor eerder te zien en vroegtijdig in te grijpen. Dit verhoogt niet alleen de veiligheid, maar ook de beschikbaarheid van het drukbezette spoor.

Inspection is een onderneming gespecialiseerd in advisering en engineering van beheer, onderhoud en conditie-monitoring van infrastructuur.

Sherloc, de meettrein van Inspection, levert een helder, accuraat en actueel beeld van de kwaliteit van wissels en sporen. Sherloc is voorzien van verschillende lasersystemen en HD-fotocamera's. Deze treinen maken drie à vier keer per jaar van elk stukje spoor een foto. Dat levert een enorme hoeveelheid beeldmateriaal op dat door inspecteurs van Inspection wordt bekeken. Dit inspectieproces is een veel veiliger en efficiënter alternatief ten opzichte van de vroegere situatie waarbij inspecteurs langs het spoor liepen.

Met behulp van deep learning hebben wij, samen met Inspection, een systeem ontwikkeld voor het automatisch detecteren van defecten in spoorstaven gebaseerd op het beeldmateriaal geproduceerd door Sherloc. In de nieuwe opzet bekijkt de inspecteur slechts die stukken spoor die door het algoritme als verdacht worden bestempeld. Dit betekent dat hij 80% van de beelden niet meer hoeft te bekijken. De overige foto's bekijkt hij nog wel, maar dan om te controleren of er daadwerkelijk van een defect sprake is en vervolgens te beoordelen welke acties nodig zijn om het eventuele defect te repareren. Men ziet beginnende defecten eerder en kan vroegtijdig ingrijpen als dat nodig is. Dit verhoogt niet alleen de veiligheid, maar vermindert ook de reizigershinder door onderhoud.

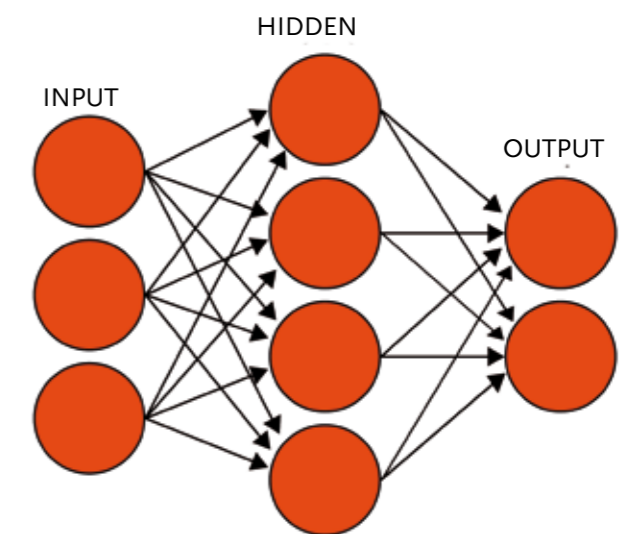
Hoe werkt deep learning?

Deep learning is een vorm van machine learning gebaseerd op neurale netwerken. Een neurale netwerk, ondanks de op biologie geïnspireerde naam, is in zijn meest eenvoudige vorm gewoon een functie $f_{\theta}: \mathbb{R}^n \rightarrow \mathbb{R}^k$ gepresenteerd door een gerichte graaf (zie figuur 1). In deze graaf heeft iedere gerichte kant a een gewicht θ_a . Het uitrekenen van deze functie (het evalueren van het neurale netwerk) werkt als volgt: de input knopen (de knopen zonder binnenkomende kanten) krijgen een waarde overeenkomstig met de n verschillende coördinaten van de input. Vervolgens worden de waarden van de overige knopen bepaald door, voor iedere knoop v , $g_v(\sum_a \theta_a x_a)$ uit te rekenen. Hier wordt gesommeerd over alle in v binnenkomende kanten a , is x_a de waarde van de knoop waar kant a vandaan komt, en zijn de g_v vooraf gedefinieerde niet-lineaire functies. Het resultaat van de functie f_{θ} is uiteindelijk af te lezen als de waarden van de output knopen (knopen zonder uitgaande kanten).

Neurale netwerken zijn er in vele verschijningsvormen. Zo kun je eindeloos variëren met de topologie van het netwerk, dus ook met het aantal parameters θ_a , en

met de deelfuncties g_v . In het voorbeeldplaatje zie je een gelaagd netwerk met één input laag, één interne (hidden) laag en één output laag. Wij gebruiken een topologie die ook wel een convolutioneel neurale netwerk genoemd wordt. De term deep learning verwijst naar neurale netwerken die diep zijn, waarmee bedoeld wordt dat ze meer dan alleen een input en een output laag hebben. Het voorbeeld is dus een heel eenvoudig deep learning model.

Een neurale netwerk wordt pas nuttig als de functie die men ermee uitrekent daadwerkelijk iets betekent. In theorie kunnen neurale netwerken iedere functie willekeurig goed benaderen, mits het netwerk groot genoeg mag zijn. Door middel van een set trainingsdata, data waarvan de gewenste uitkomst van het netwerk bekend is, wordt gezocht naar waarden voor de parameters zodanig dat het model deze data benadert, en daarnaast op nieuwe ongeziene data ook gewenste uitkomsten geeft. Dit is niet heel anders dan voor statistische modellen,



Figuur 1. Voorbeeld van een neurale netwerk

echter met een aantal belangrijke verschillen:

- Ten eerste is het vaak niet mogelijk en praktisch niet relevant om de best fittende parameters te vinden, zoals bij maximum likelihood, omdat het bijbehorende optimalisatie probleem niet convex is en daarnaast ook heel hoog dimensionaal. Sterker nog, om overfitting te voorkomen, wil je het optimum niet eens vinden.
- Ten tweede is het aantal parameters vaak groter (soms wel ordes groter) dan het aantal punten in de trainingsdata. Hierdoor is het optimale netwerk dus zelfs niet uniek bepaald.
- Ten slotte kijken we meer naar het neurale netwerk als *black box* waarbij we tevreden zijn als de uitkomsten op zowel de trainingsdata als de niet eerder geziene data goed genoeg zijn, zonder precies te willen begrijpen waarom juist deze model parameters werken.

Door implementaties op krachtige grafische kaarten die veel meer rekenkracht hebben dan moderne processoren is het mogelijk neurale netwerken van significante omvang op niet al te dure hardware te trainen. Wij gebruiken hier een Titan X grafische kaart van Nvidia voor en het TensorFlow framework van Google.

Tot zover de theorie, nu de toepassing.

Sherloc levert een film van het spoor. Als je de beelden van deze film achter elkaar legt ontstaat een hele grote langwerpige foto van het spoor. Om deze beelden geschikt te maken voor classificatie met een neurale netwerk gaan we de film eerst omzetten in losse plaatjes van een kleiner, vast formaat: de inputs x van het neurale netwerk. Zie de resulterende plaatjes in figuur 2. Vervolgens zorgen we ervoor dat het netwerk een getal tussen 0 en 1 als output geeft dat te interpreteren is als de kans dat dit specifieke plaatje een defect bevat.

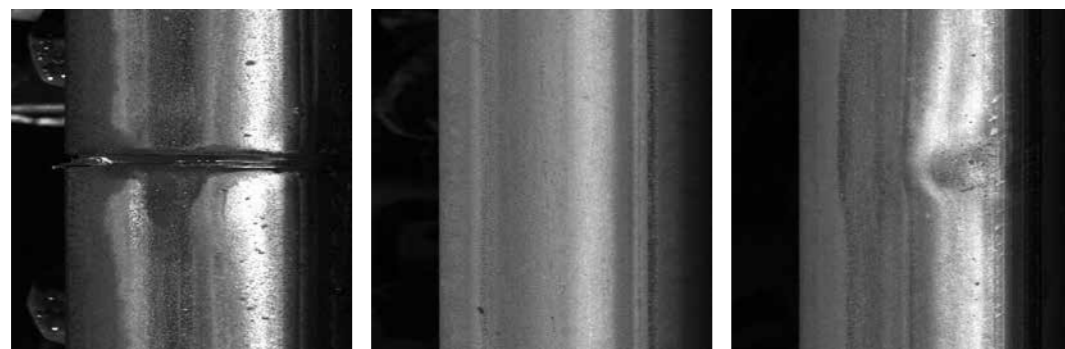
Omdat de film ongeveer 3 keer zo breed is als de spoorstaaf, en deze spoorstaaf niet altijd op dezelfde plek

in de film ligt, zoeken we eerst de spoorstaaf op. Dit doen we door per frame, per kolom van pixels in het frame, een histogram van de grijswaarden te maken. Uit deze histogrammen (en de locatie van de spoorstaaf in het vorige frame) is op te maken waar de spoorstaaf ligt. Hierna, kunnen we een venster (*sliding window*) over de spoorstaaf schuiven, en om de zoveel pixels een plaatje maken.

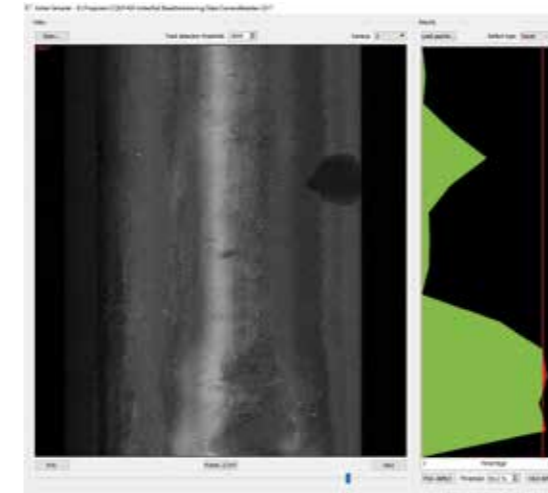
Om een verzameling trainingsdata te maken, worden films handmatig door inspecteurs geannoteerd met waar de defecten zich bevinden. Dit is te vertalen naar of een plaatje binnen het sliding window een defect bevat of niet. Als het netwerk uiteindelijk gebruikt wordt om defecten in het spoor te detecteren dan levert dit per verschuiving van het venster een kans op een defect op. Dit visualiseren we naast het spoor als een grafiek met op de verticale as de afstand over het spoor en op de horizontale as de kans op een defect, zie figuur 3.

De gebruiker kan vervolgens kiezen bij welke kans hij een defect wil zien en zo de *trade-off* tussen de hoeveelheid werk en de kans dat er defecten gemist worden beïnvloeden. Hierbij is het van groot belang dat het aantal *false negatives* (wel een defect, niet gedetecteerd) klein is, omdat de inspecteur deze dus niet meer ziet, terwijl *false positives* (geen defect, wel als dusdanig gedetecteerd) niet zo erg zijn. Gelukkig voor de staat van ons spoor-net speelt hierbij de extra complicatie dat we juist heel veel beelden van schoon spoor hebben terwijl het aantal beelden van defecten vele ordes van grootte kleiner is. Hiermee houden we expliciet rekening met het samenstellen van de trainingsdata en tijdens het trainen straffen we fouten op het classificeren van defecten veel harder af dan classificatiefouten op schoon spoor.

Een van de grootste uitdagingen van dit project was het zodanig samenstellen van de trainingsdata dat hierin zo veel mogelijk van alle natuurlijke variatie die in de praktijk voorkomt terugkomt. Zo moet het netwerk leren dat *squats* wel defecten zijn, maar tegelijkertijd dat regen-druppels en vetvlekken dit niet zijn. Daarnaast kan het



Figuur 2. Links een stuk spoor met een ES-las (geen defect), in het midden schoon spoor, en rechts een squat categorie B (een defect)



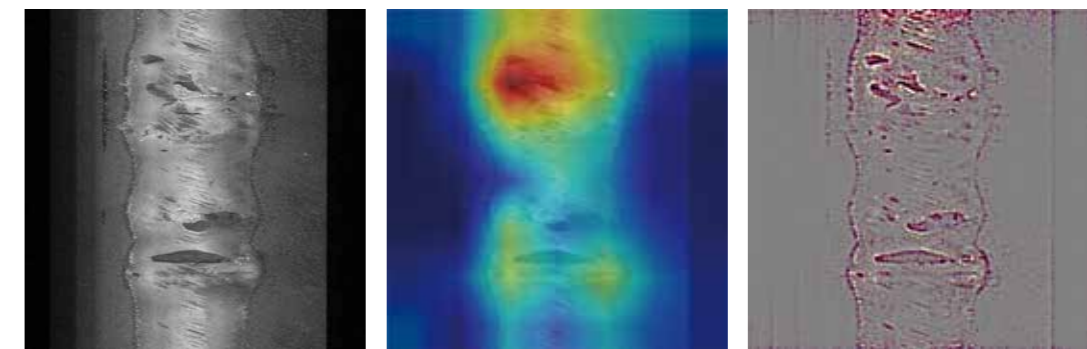
Figuur 3. Visualisatie van de kans op een defect

spoor er in verschillende omstandigheden, zoals bij laagstaande zon, ineens heel anders uit zien.

Ten slotte, hebben we om het vertrouwen in het netwerk te vergroten geëxperimenteerd met verschillende technieken om zichtbaar te maken op welke onderdelen van een input plaatje het neurale netwerk zijn beslissing baseert, zie figuur 4. Dit is essentieel om inzicht te krijgen in de werking van de black box en om vertrouwen in het systeem te kweken bij inspecteurs.

Conclusie

Automatische spoorinspectie met behulp van deep-learningtechnieken is een voorbeeld van hoe data science in de praktijk kan werken. CQM ontwikkelde samen met Inspectionation een zelflerend systeem voor het automatisch detecteren van defecten op het spoor. In de toekomst biedt de huidige oplossing mogelijkheden om de meest bereden kritische sporen nog vaker te inspecteren en de kwaliteit nog verder verhogen. Verder kan de oplossing ook worden ingezet op het buitenlandse spoor-netwerk of metro- en tramnetwerken. Daarnaast is de onderliggende techniek geschikt om toe te passen voor het inspecteren van wegen, bruggen en tunnels.



Figuur 4. Links een stuk spoor met overduidelijke defecten; in het midden een heatmap van regio's in de afbeelding waarop het neurale netwerk zijn beslissing baseert; rechts een meer gedetailleerde weergave van de belangrijkste pixels. (Naar Selvaraju et al., 2017)

NOTEN

1. Inspectionation, voortgekomen uit onderhoudsaannemer VolkerRail, is een onderneming gespecialiseerd in advisering en engineering van beheer, onderhoud en conditie monitoring van infrastructuur.
2. CQM (Consultants in Quantitative Methods) is een adviesbureau op het gebied van logistieke optimalisatie, industriële statistiek en data science gevestigd in Eindhoven.
3. De Hendrik Lorentz Prijs is bedoeld voor een organisatie binnen het bedrijfsleven of de overheid die op een onderscheidende en innovatieve manier data science toepast. <https://nederlandsedatascienceprijsen.nl>

LITERATUUR

- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. Cambridge MA: MIT Press [een gedegen inleiding in het vakgebied].
- Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2017). *Grad-CAM: Gradient-weighted Class Activation Mapping*. ICCV

JOHAN VAN ROOIJ is als Senior Consultant werkzaam bij CQM. Daarnaast heeft hij een aanstelling als universitair docent aan het Departement Informatica aan de Universiteit Utrecht. E-mail: Johan.vanRooij@cqm.nl

HUUB VAN DEN BROEK is als Senior Consultant werkzaam bij CQM en is projectleider van dit project. E-mail: Huib.vandenBroek@cqm.nl