

Flood recognition with pixels segmentation



Ruben Rajan
1682750

Inleiding

Dit rapport beschrijft hoe een model voor een toepassing kan worden gecreëerd om overstromingsgebieden op luchtfoto's te herkennen. De visuele taak van dit model is om te bepalen waar overstroming op een afbeelding voorkomt. Om te bepalen welke delen van de afbeelding wel en niet bij de overstroming horen, wordt elke pixel individueel geanalyseerd. Voor elke pixel wordt bepaald of deze tot de overstroming behoort of niet. De software genereert op basis van een afbeelding een masker dat over het originele beeld kan worden gelegd. Dit masker geeft voor elke pixel aan of deze al dan niet deel uitmaakt van de overstroming.

Literatuur

In verschillende literatuurbronnen wordt voor beeldsegmentatie vaak gebruikgemaakt van K-means clustering. K-means is een ongesuperviseerde machine learning-methode waarbij gegevens worden gegroepeerd in clusters. Tijdens het K-means proces wordt het centrum van elk cluster berekend totdat het gewenste aantal clusters is bereikt. Een voorbeeld van K-means clustering is te vinden in de tutorial "Color Quantization using K-Means" op de website van scikit-learn. Deze tutorial past K-means clustering toe om kleuren in een afbeelding in een beperkt aantal clusters in te delen. Hiermee kan het aantal kleuren in een afbeelding bijvoorbeeld worden teruggebracht van 256 kleuren naar 10 kleuren. Deze techniek staat bekend als kleurcompressie. In ons experiment gebruiken we kleurcompressie om het kleurbereik van water in elke afzonderlijke afbeelding te bepalen.

Methode

Het experiment bestaat uit drie hoofdonderdelen:

1. Het eerste model maakt gebruik van een neurale netwerk om te bepalen of elke pixel bij de overstrooming hoort. Het model bevat conventionele en dense lagen. De input voor het neurale netwerk is een grayscale patch van de afbeelding, en we hebben gevarieerd in de grootte van deze patches.
2. Het tweede model converteert de afbeelding eerst naar grayscale en past vervolgens K-means clustering toe. Elk cluster krijgt zijn eigen numpy-dimensie, waarbij elke pixel binnen dat cluster de waarde 1 krijgt, en alle andere pixels krijgen de waarde 0. We voeren erosie uit op elke dimensie en behouden de dimensie met de meeste resterende pixels als masker voor onze voorspellingen. Door erosie te gebruiken, bepalen we welk deel van de afbeelding het grootste en meest homogene vlak vertegenwoordigt, dat we aannemen als de overstrooming.
3. In het derde experiment combineren we de twee voorgaande methoden door het masker van het tweede model als een extra dimensie toe te voegen aan het eerste model, waarna we het neurale netwerk trainen.

Het experiment

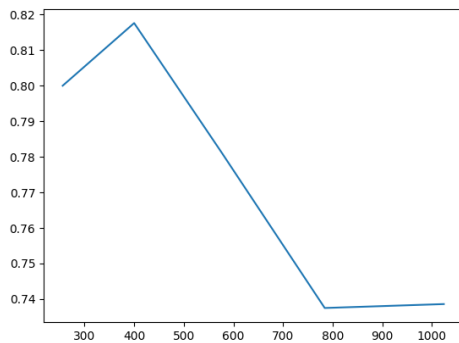
Voor ons eerste model hebben we de afbeeldingen verkleind naar 250 bij 250 pixels en patches genomen met stappen van 32 pixels. Deze stappen waren nodig vanwege beperkte rekenkracht, aangezien het experiment werd uitgevoerd op een laptop zonder GPU. We begonnen met patches van 16 bij 16 pixels en verhoogden deze tot 32 bij 32 pixels.

Het tweede experiment gebruikte onverkleinde afbeeldingen, en we varieerden het aantal clusters bij K-means clustering.

In het derde experiment hebben we de modellen geselecteerd met de optimale testnauwkeurigheid en ze gecombineerd. Het masker van het tweede model werd toegevoegd als een extra dimensie aan de afbeelding, waarna we 2D-patches maakten.

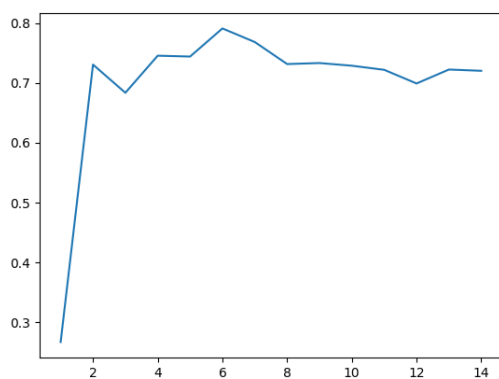
Resultaten

In figuur 1, 2 en 3 worden de resultaten van de experimenten getoond. In figuur 1 vertegenwoordigt de x-as het totale aantal pixels in de patch. De piek in het eerste experiment wordt bereikt bij een patch van 20 bij 20 pixels met een nauwkeurigheid van 81 procent. In het tweede experiment wordt een piek van 80 procent bereikt bij K-means clustering met 6 clusters.



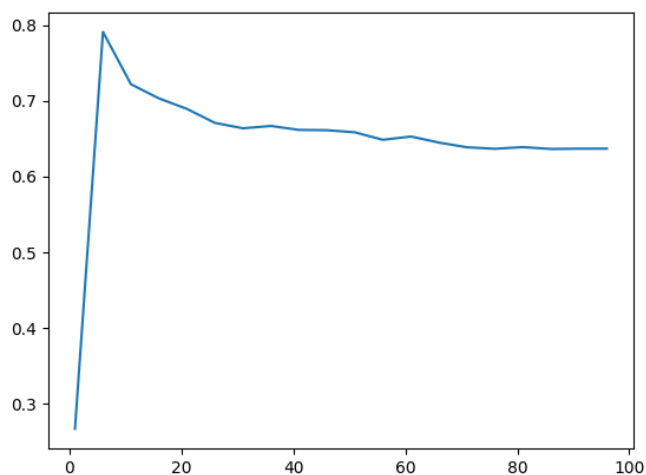
x-as: aantal pixels in patch
y-as: nauwkeurigheid

Figuur 1



x-as: aantal kmean clusters
y-as: nauwkeurigheid

Figuur 2



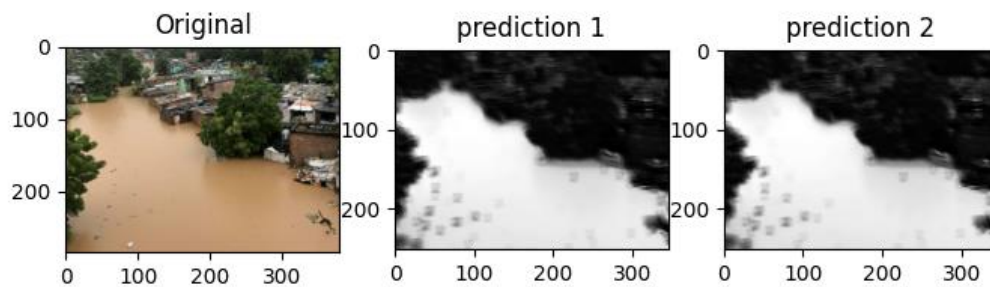
x-as: aantal kmean clusters
y-as: nauwkeurigheid

Figuur 3

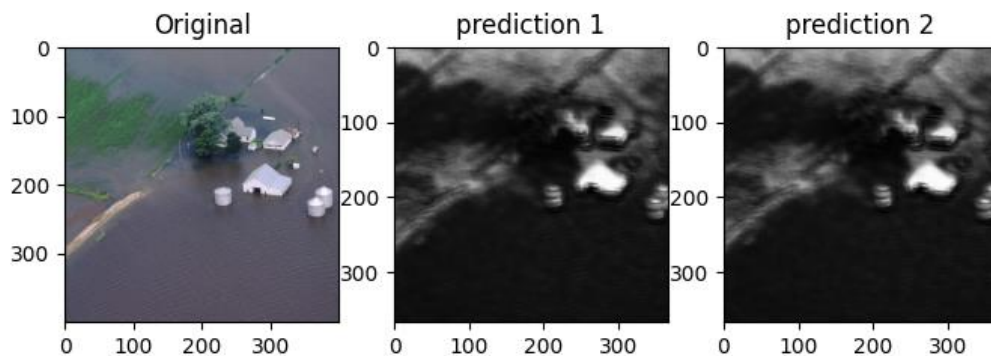
De piek bij het eerste experiment is bij een patch van 20 bij 20 pixels met 81 procent.
De piek bij het tweede experiment is 80 procent met kmean klustering van 6 klusters.

Voorbeeld afbeeldingen

Figuur 4 en 5 tonen voorbeeldafbeeldingen. Een witte pixel geeft aan dat het model voorspelt dat het een overstromingspixel is. Prediction 1 is het resultaat van het eerste model met het neurale netwerk, terwijl Prediction 2 het resultaat is van K-means clustering en erosie. Figuur 4 toont geen golven, terwijl figuur 5 een minder scherpe afbeelding met golven bevat.



Figuur 4



Figuur 5

Experiment 3

Voor experiment 3 hebben we een model gemaakt met een patchgrootte van 20 bij 20 pixels en een extra laag waarbij het masker is gemaakt op basis van K-means clustering met 6 clusters. De nauwkeurigheid van dit model bedraagt 75 procent.

Conclusie

Dit experiment laat zien dat een model voor het herkennen van overstromingen op luchtfoto's kan worden gemaakt met zowel een neurale netwerk als K-means clustering met erosie. Beide methoden vertonen vergelijkbare nauwkeurigheid. Het combineren van deze methoden lijkt in dit experiment geen significant effect te hebben en kan zelfs leiden tot een lichte afname in nauwkeurigheid. Het is belangrijk op te merken dat beide modellen goed presteren op scherpe afbeeldingen zonder golven, zoals te zien is in figuur 4 en 5.

Discussie

Dit experiment toont wat er mogelijk is, maar we moeten er rekening mee houden dat de nauwkeurigheidscijfers niet optimaal zijn vanwege beperkte rekenkracht en beperkte optimalisatie van hyperparameters en neurale netwerkstructuren. In toekomstige experimenten kan het interessant zijn om te onderzoeken hoeveel procent van de overstromingsinformatie verloren gaat bij de toepassing van erosie in plaats van hoeveel procent behouden blijft.

Literatuurlijst

Bibliography

A. Bannari, A. G.-B. (2017). Detection of Areas Associated with Flash Floods and Erosion Caused by Rainfall Storm Using Topographic Attributes, Hydrologic Indices, and GIS. *Global Changes and Natural Disaster Management: Geo-information Technologies*, 155-174.

Color Quantization using K-Means. (2023). Opgehaald van scikit-learn: https://scikit-learn.org/stable/auto_examples/cluster/plot_color_quantization.html

Pedregosa, F. a. (2011). Scikit-learn: Machine Learning in {P}ython. *Journal of Machine Learning Research*, 12.

MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. In L. M. Le Cam & J. Neyman (Eds.), *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability* (Vol. 1, pp. 281–297). California: University of California Press.

Lloyd, S. P. (1957). Least squares quantization in PCM. Technical Report RR-5497, Bell Lab, September 1957.