

Flood recognition with pixels segmentation



Ruben Rajan
1682750

Inleiding

In dit rapport wordt beschreven hoe een model voor een applicatie kan gemaakt worden voor het herkennen van een overstrooming `s gebied in een luchtfoto.

De vision taak waar een model voor is ontwikkeld, is om te kijken waar de overstrooming op een afbeelding is. Om te bepalen welke gedeeltes wel en niet bij een overstrooming horen wordt per pixel gekeken. Voor elke pixel wordt bepaald als het wel of niet bij een overstrooming hoort.

De software moet op basis van een image dus een masker maken die over het originele plaatje kan zetten. Het masker geeft per pixel aan als het wel of geen overstrooming pixel is.

Literatuur

In verschillende literatuur wordt voor segmentatie vision taken kmean klustering gebruikt. Kmean is een unsupervised manier van machine learning waarbij data wordt ingedeeld in clusters. Bij kmean wordt het middelpunt berekend tot er gewenste clusters zijn. In de tutorial Color Quantization using K-Means (van de website van sklearn) wordt kmean toegepast. In de tutorial wordt kmean klustering gebruikt om de kleuren in een x aantal clusters in te delen. Door de kleuren in clusters in te delen, kan je het aantal kleuren van bijvoorbeeld 256 kleuren terugbrengen naar 10 kleuren. In de tutorial wordt dus gedaan aan kleur compressie gedaan. In het experiment wordt kleur compressie gebruikt om te kijken, in welk kleur bereik het water zit per afbeelding.

Methode

Het experiment bestaat uit twee onderdelen

Het 1e model bestaat uit door een neurale netwerk dat kan herkennen als een pixel wel of niet bij de overstrooming hoort. Het model bestaat uit conventionele en dense layers. De input van het neurale netwerk is een grayscale patch van de image. In het experiment wordt gekeken naar verschillende patch grote. De patch is in de vorm van een vierkant.

In het 2^e model wordt de image eerst naar grayscale omgezet waarna een kmean klustering wordt gebruikt. Na kmean krijgt elke cluster zijn eigen numpy dimensie. Waarbij elke pixel binnen de cluster de waarde 1 krijgt en de rest waarde 0. Elke dimensie wordt geërodeerd. Na het eroderen wordt de dimensie die de meeste pixel overhoudt bewaard en als masker gebruikt voor de voorspellingen. Door het eroderen wordt dus bepaald wat het grootste egale vlak is. Met deze methode gaat er van uit dat de grootste egale vlak dus het water is.

In het 3^e experiment worden de 2^e methodes gecombineerd door de masker van het 2^e model toe te voegen als extra dimensie aan het 1^e model waarna het neurale netwerk wordt getraind.

Het experiment

De data komt van een dataset van kaggle. Tijdens het experiment voor het eerste model zijn de afbeeldingen verkleind naar 250 bij 250 pixels. De patches van de pixels werden met stappen van 32 pixel uit de afbeelding genomen. De afbeelding verkleinen en patches met stappen nemen werden gedaan om de tijd in te korten van het draaien van de code. Aangezien het experiment op een laptop werd gedraaid zonder GPU moesten helaas deze stappen genomen worden.

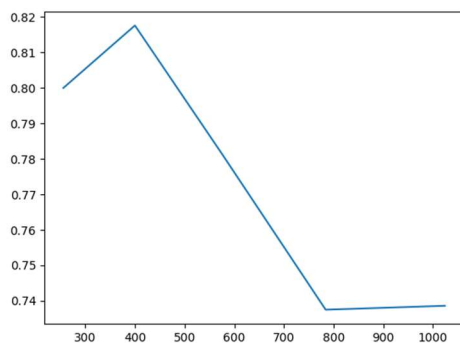
De patch grote begon met 16 bij 16 pixels en werd verhoogd met 4 bij 4 pixels tot 32 bij 32 pixels. Na 6 verschillende modellen te trainen werden op andere foto's dan de trainingsdata een voorspelling masker gemaakt. Daarna werd gekeken hoeveel procent het voorspellingsmasker met het originele masker overeenkomt.

Bij de 2^e experiment werden de afbeeldingen niet verkleind. In het 2^e model is geëxperimenteerd met het aantal clusters in de kmean klustering.

In het 3^e experiment zijn de modellen met de paramaters met de optimale accuraatheid bij het testen uitgekozen en samengevoegd. Het voorspellingsmasker van het 2^e model is als extra dimensie toegevoegd aan de afbeelding waarna 2-dimensionale patches zijn gemaakt.

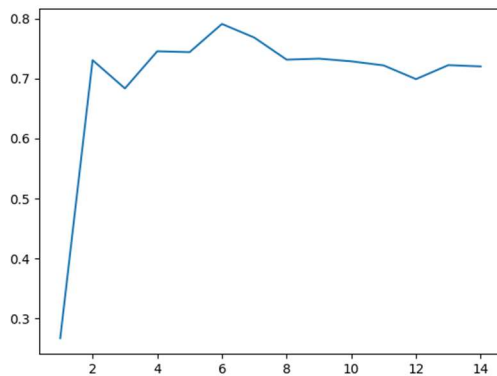
Resultaten

In figuur 1, 2 en 3 zijn de meet resultaten van de experimenten te zien. In figuur 1 zijn de resultaten van experiment 1 te zien. De x as bij figuur 1 is het totaal aantal pixels van de patch, dus een patch van 20 bij 20 pixels heeft 400 pixels totaal. Figuur 2 en 3 zijn de resultaten van experiment 2 te zien.



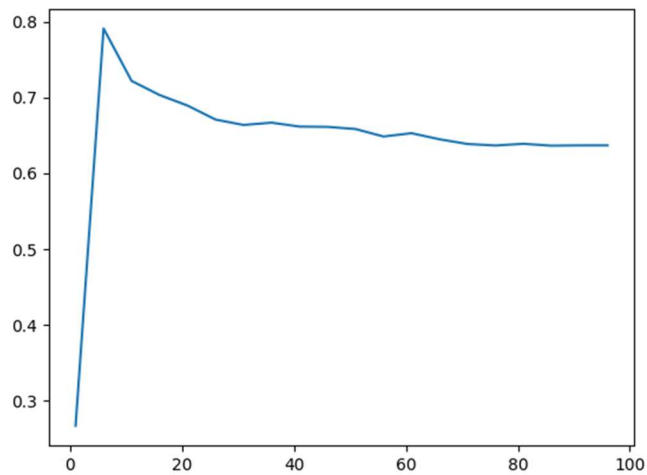
x-as: aantal pixels in patch
y-as: nauwkeurigheid

Figuur 1



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

Figuur 2



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

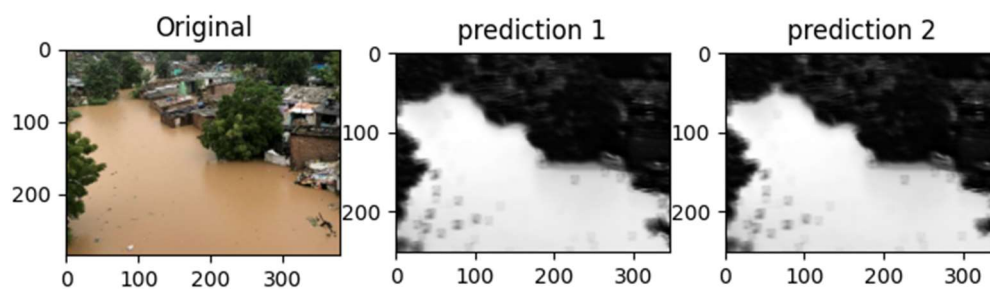
Figuur 3

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.

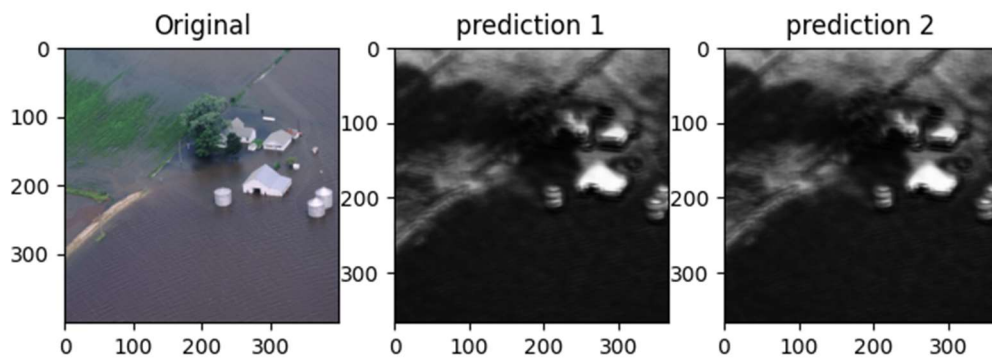
Voorbeeld afbeeldingen

In figuur 4 en 5 zijn voorbeeld afbeeldingen te zien. Een wit pixel betekent dat het model een overstromings pixel voorspelt. Prediction 1 is het resultaat van het 1^e model met neuraal netwerk en prediction 2 is het resultaat met kmean clustering en eroderen.

Figuur 4 bevat geen golven. Figuur 5 is een minder scherpe foto met golven.



Figuur 4



Figuur 5

Experiment 3

Voor experiment 3 waarbij de een model is gemaakt met een patch grote van 20 bij 20 pixels en een extra laag heeft waarbij de mask is gemaakt gebaseerd op een kmean klustering van 6 clusters.

De nauwkeurigheid van dit model is 75 procent.

Conclusie

Het experiment toont aan om een model voor een applicatie om te herken waar de overstroming is op een luchtfoto kan gemaakt worden met zowel een neuraal netwerk als met kmean clustering en eroderen. De 2 methodes zijn ongeveer even nauwkeurig. Het combineren lijkt met de parameters in dit experiment geen effect te hebben. Het model is door te combineren zelf een klein beetje minder nauwkeurig. Alle 2 de modellen werken alleen wel goed op foto's die scherp zijn en zonder golven is te zien in figuur 4 en 5.

Discussie

Dit experiment toont aan wat mogelijk is. Omdat er gekort is in rekenkracht zijn de nauwkeurigheid cijfers niet betrouwbaar. Het experiment laakt dus zien wat er minimaal mogelijk is. Ook zijn er veel meer hyper parameters beschikbaar en is er niet gefocust op de layers van het neuraal netwerk. Bij experiment 2 kan in het vervolg ook gekeken hoeveel procent verdwijnt in plaats van hoeveel procent overblijft.

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.