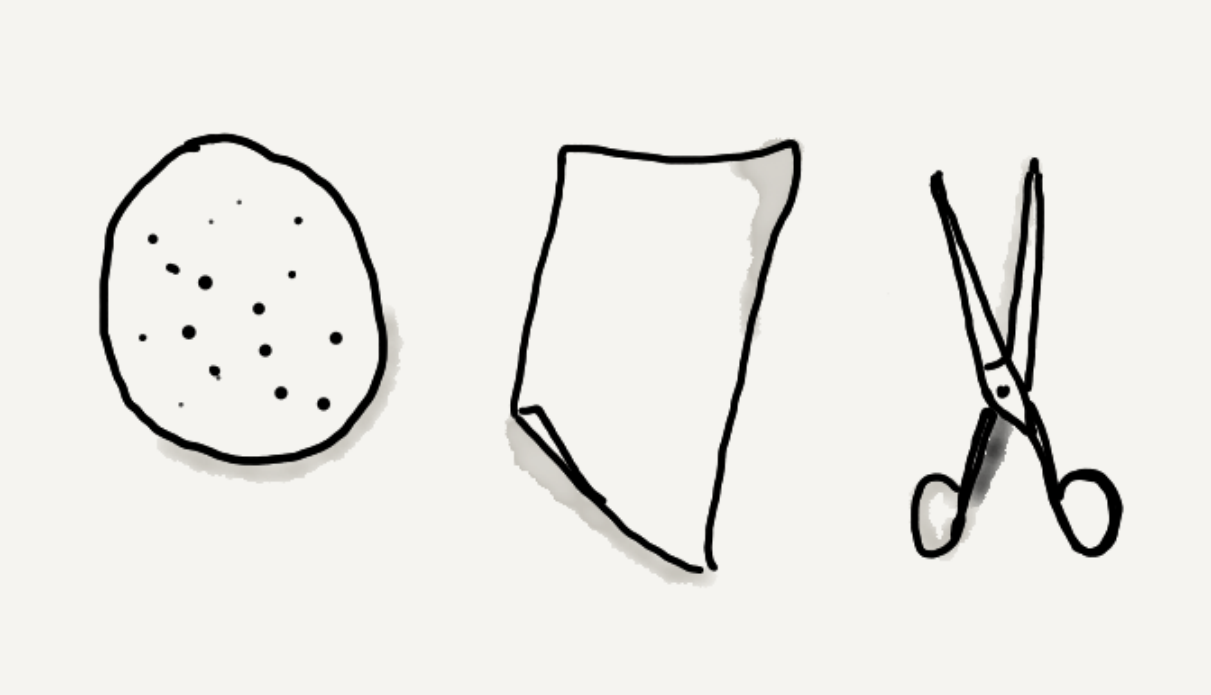
ComputerVision

Steen Papier Schaar



Datum: 27-3-2021

Door: Koen Brave

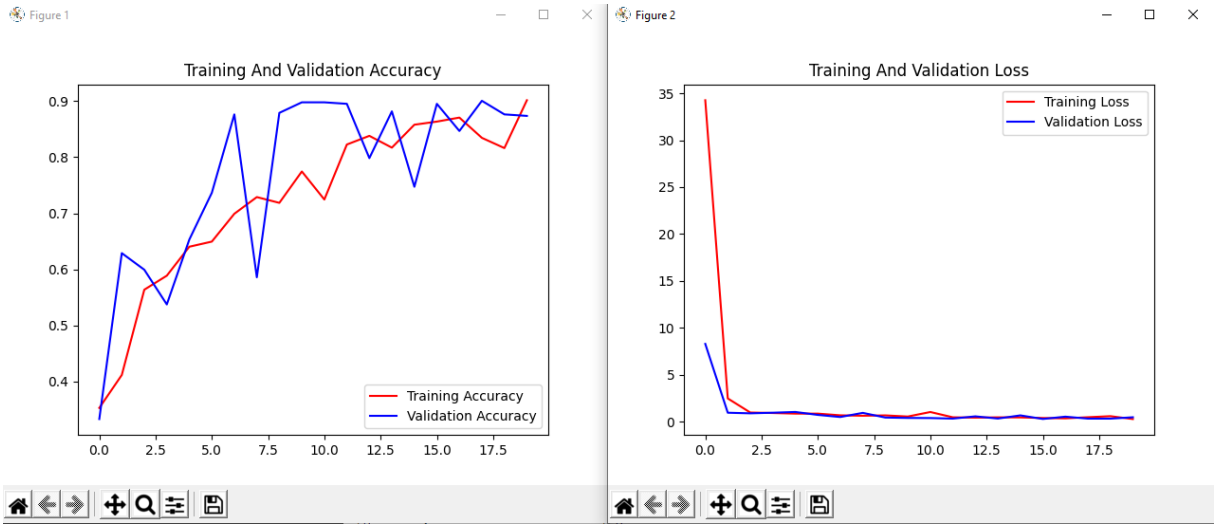
Paper: https://aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16386/16414

De bedoeling is om 3 soorten vormen van handen te herkennen. De eerste is steen dat zou dus een vuist zijn. De tweede is papier dat zijn alle vingers uitgestoken een platte hand en als laatste schaar waar bij 2 vingers zijn uitgestoken.

Ik ben begonnen met de training data inladen van uit het mapje (/Rock-Paper-Scissors/train/). Dit dan voor alle 3 de classes. In het paper word vertelt dat het handig het aantal pixels te verkleinen naar 64x64(punt 7), dat heb ik gedaan met de Conv2D filter. Op het moment dat dat ik 32x32 weglaat kreeg ik een input error dat de images niet goed waren. Dit heeft waarschijnlijk te maken met de grote van de afbeeldingen. MaxPooling2D word gebruikt om de afbeelding correct te verkleinen, hierbij heb ik 2x2 gekozen en het grootste getal gepakt. Na die filters word de filter flatten uitgevoerd, dit is van belang om de input klein te houden en layers bij elkaar te voegen. In de paper staat dat je goed moet opletten voor overfilteren(punt 9). Na wat goed zoeken op het internet staat dat dropout een goede filter is om toe te voegen. Na wat experimenteren ben ik op het getal 0.4 uitgekomen die mijn het beste resultaat levert met deze dataset.

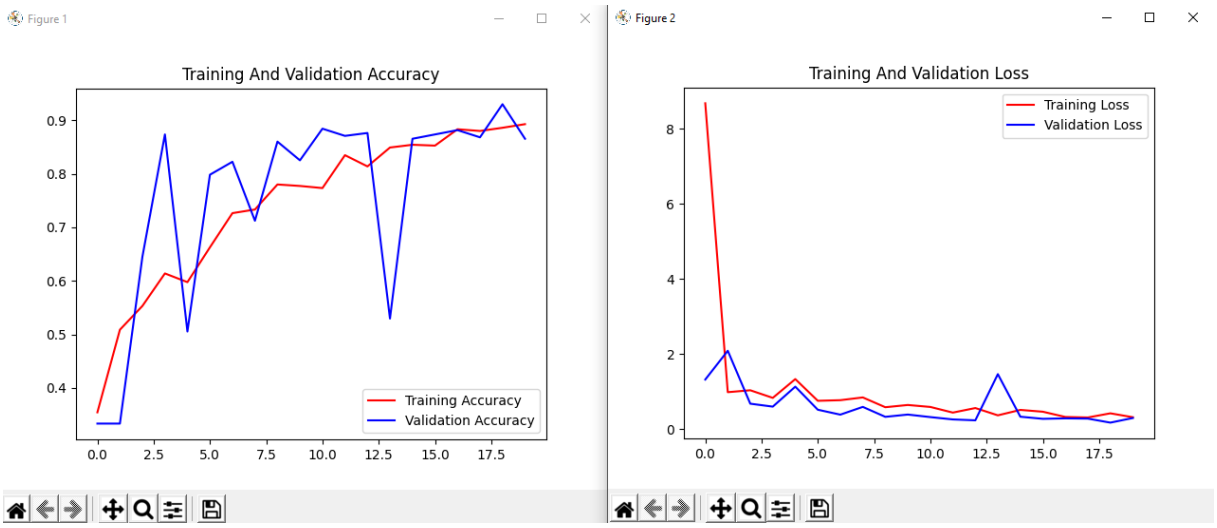
De eerste bestaat uit 32 kleur, maxpooling 2,2, flatten, dense 512 met relu en dense 3 softmax

Epochs 20 met steps 20.



De eerste test was een groot succes met wel een accurate van 90%

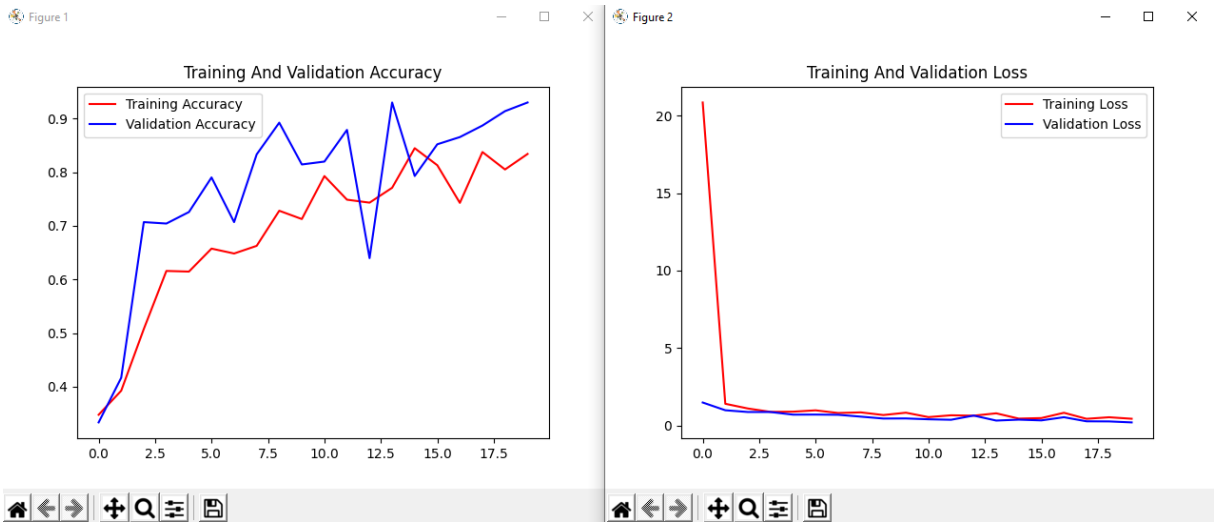
Tweede Test bestaat uit 32 kleuren en 64 kleuren, maxpooling 2,2, flatten, dense 512 met relu en dense 3 softmax, Epochs 20 met steps 20. Het grote verschil dit ik 64x heb wat in het paper werd aangeraden.



Grappig genoeg is met 32 en 64 de accuratie 89% wat dus betekend dat 32 op het eerste gezicht beter is. Je kan ook in de grafiek zien bij figuur 1 dat er een grote piek naar beneden is waar die in test 1 veel minder erg is.

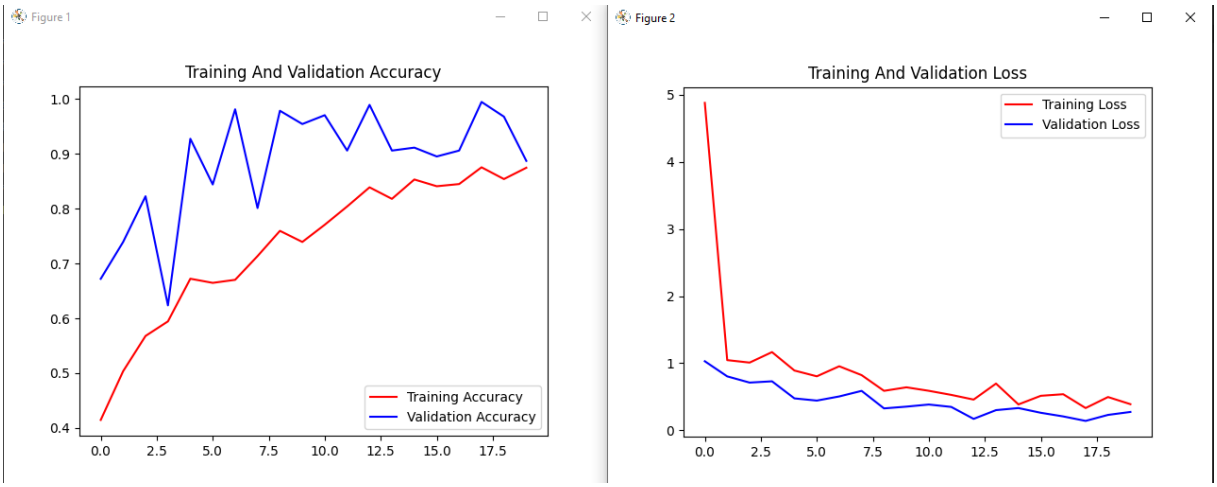
Bij de volgende 2 testen ga ik weer precies het zelfde doen met kleuren maar dit keer niet dense van 512 maar een dense van 256 om te kijken of dit een groot verschil levert.

Test 3:



Het resultaat is een stuk naar beneden gegaan naar 83% dit is een erg grote daling maar bij de volgende test met de 64 kleuren er bij gebeurt er wat verrassends.

Test 4:



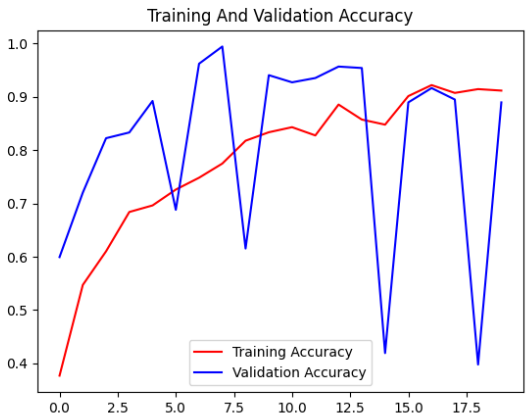
Wat heel erg verrassend is is dat de accuratie 87% is. Het betekend nogsteeds dat het omlaag is gegaan met een stuk minder dan bij test 3.

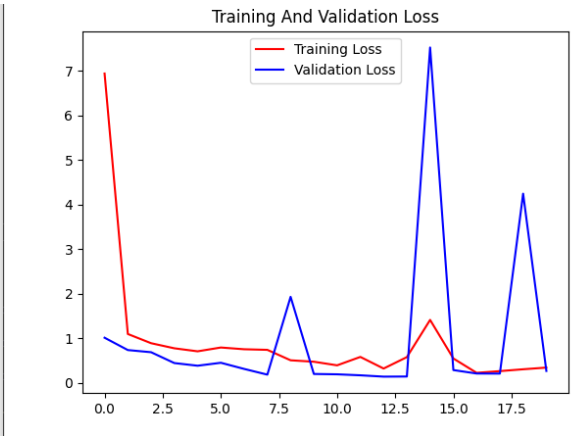
In de paper word ook aangegeven om een dropout te gebruiken. Dit heb ik ook toegepast.

Met het testen heb ik besloten om 32 en 64 te gebruiken met een dense 512 soft.

Test 5 heeft een dense 0.4.

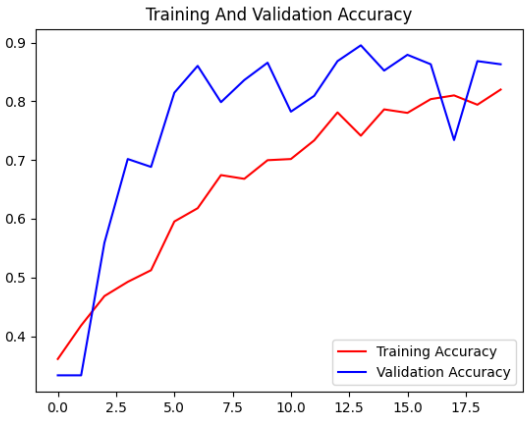
Test 5:

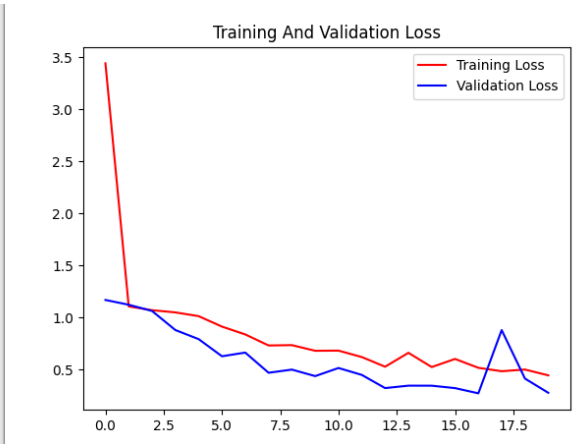




Door een dropout te gebruiken van 0.4 ben ik op een 91% accuratie gekomen! Maar om te kijken wat het verschil is heb ik ook andere dropouts getest.

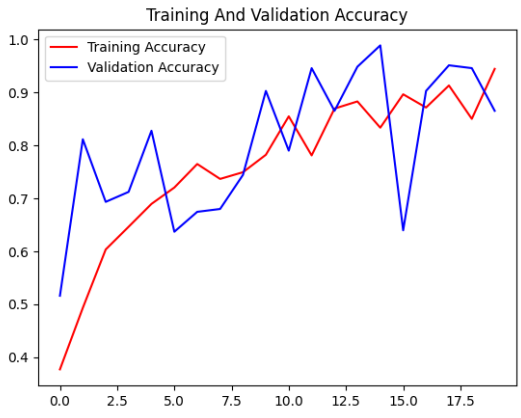
Test 6 dropout van 0.1:

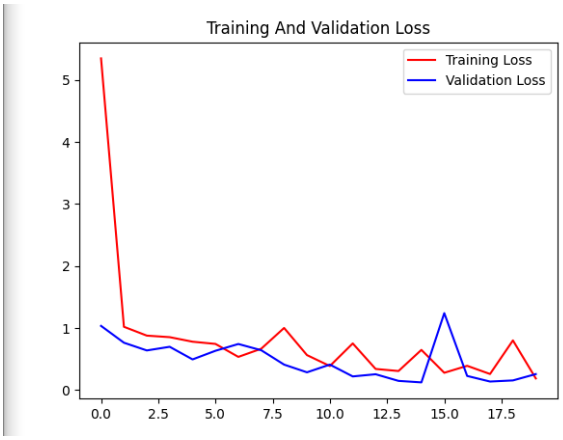




Een drop van 0.1 gaf een accuratie van maar 81%. Wat erg tegenviel als je kijkt naar het vorige resultaat.

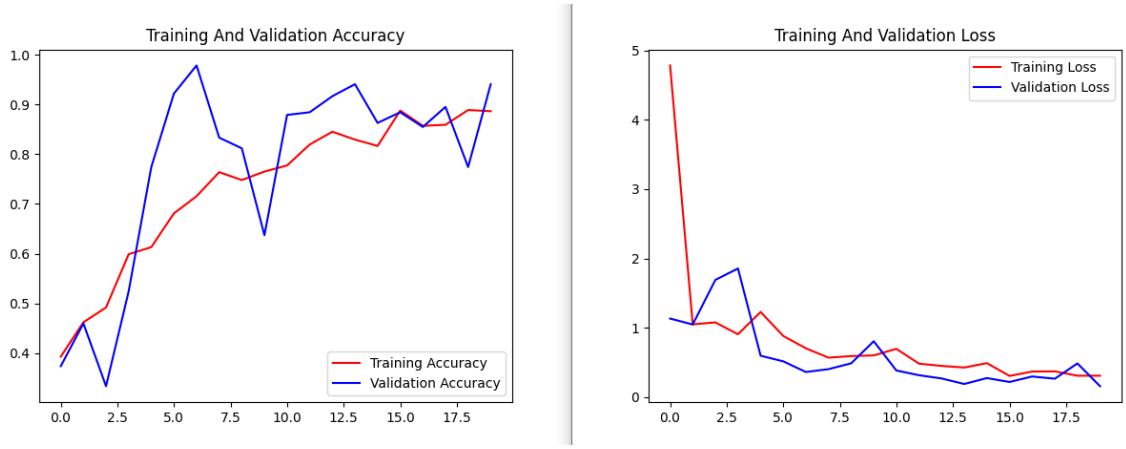
Test 7 dropout 0.7:





Een resultaat van 94% dit was nog hoger dan ik had verwacht. Na het lezen van de paper dacht ik dat het optimale 0.4 was maar 0.5 is duidelijk veel beter.

Test 8 dropout 0.8:



Dit viel erg tegen met een accuratie van 88%. Het is duidelijk dat 0.5 het beste resultaat heeft gegeven.

```
model = keras.Sequential([
    keras.layers.Conv2D(32, (3,3), activation="relu", input_shape=(150,150,3)),
    keras.layers.MaxPooling2D(2,2),

    keras.layers.Conv2D(64, (3,3), activation="relu"),
    keras.layers.MaxPooling2D(2,2),


    keras.layers.Flatten(),
    keras.layers.Dropout(0.5),
    keras.layers.Dense(512, activation="relu"),
    keras.layers.Dense(3, activation="softmax")
])
```

Dit is het model dat me uiteindelijk het beste resultaat heeft gebracht. Ook heb ik in mijn code toegevoegd dat het model word opgeslagen voor als ik het later wilde testen. Dit kan ik helaas niet naar Github uploaden want de modellen zijn te groot. Ook heb ik de history opgeslagen doormiddel van pickle zodat ik de grafieken achter kan genereren.