Eindopdracht Beeldverwerking en Computer Vision

COVID-19 face mask detector

Inhoudsopgave

Inleiding

Vanaf 1 december 2020 is het dragen van een mondkapje voor iedereen vanaf 13 jaar verplicht in het openbaar vervoer en publieke binnenruimtes. Het dragen van een mondkapje beschermt ons en anderen tegen de verspreiding van het coronavirus. Het is een van de gezonde gewoonten die ons samen beschermen tegen het virus. Daarom is het van belang dat iedereen aan de regels houd en mondkapje draagt. Om te kunnen controleren als mensen mondkapje dragen hebben we een systeem nodig. Face mask detector is de onderwerp die ik heb voor de eindopdracht van het vak vision. Dit systeem controleert als de mensen (op foto's) mondkapje dragen. Dit system kan handig wordt gebruikt op het station, op school, in de bibliotheek of in de trein.

Dataset

Dataset die ik heb gebruikt om face mask detector te trainen wordt door Prajna Bhandary gemaakt. De dataset bestaat uit 1376 foto's (690 met mondkapjes, 686 zonder mondkapjes).
Zij heeft de facial landmarks gedetecteerd op foto's voor mensen zonder mondkapje en dan met een python script wordt de foto van de mondkapje op de kin en neus gezet.

Link naar de dataset: [observations/experiements at master · prajnasb/observations (github.com)]

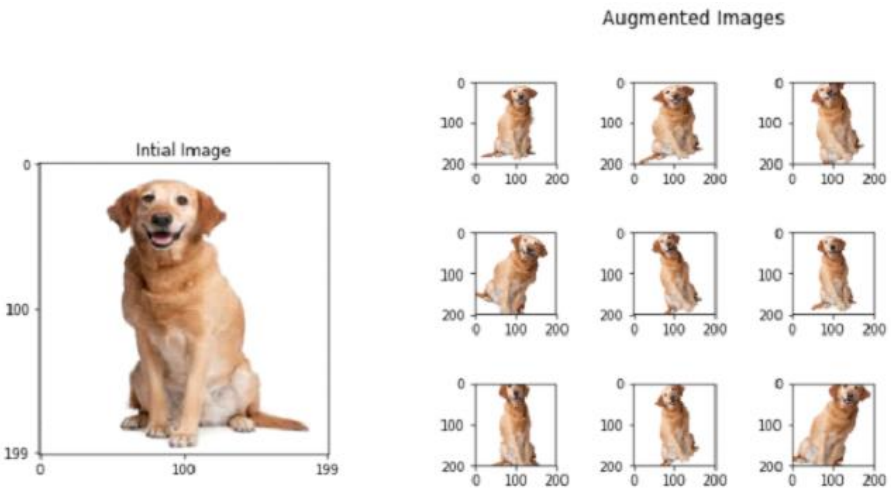
Code

- ### Training dataset maken

  Dat wordt met de create_training_data() gedaan.

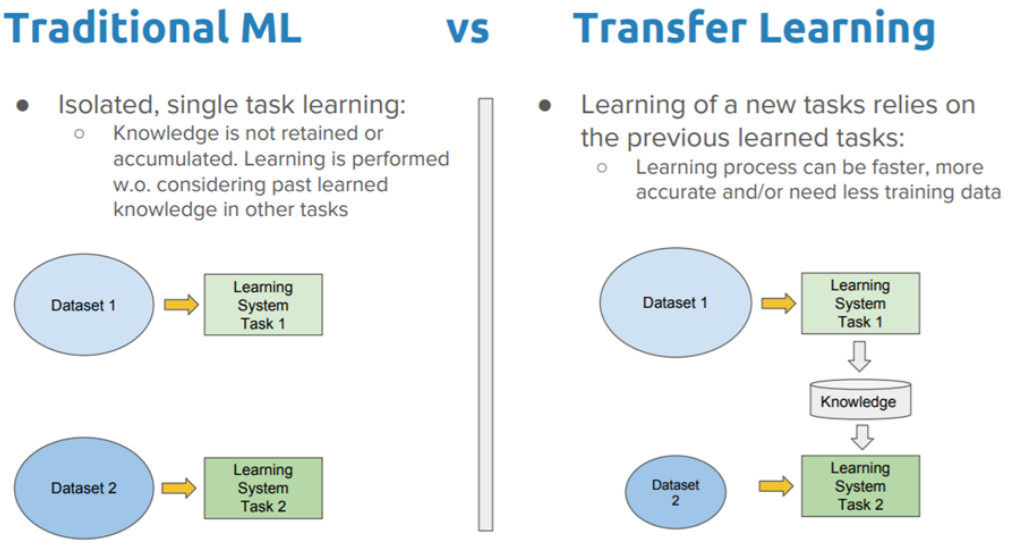
- ### Training dataset voorbereiden om kunnen te trainen

1. train_test_split: Ik heb de data gesplitst, 10% voor testen en 90% voor trainen.
2. ImageDataGenerator: ik heb keras ImageDataGenerator() functie gebruikt om mijn dataset uit te breiden en mijn model beter te kunnen trainen.

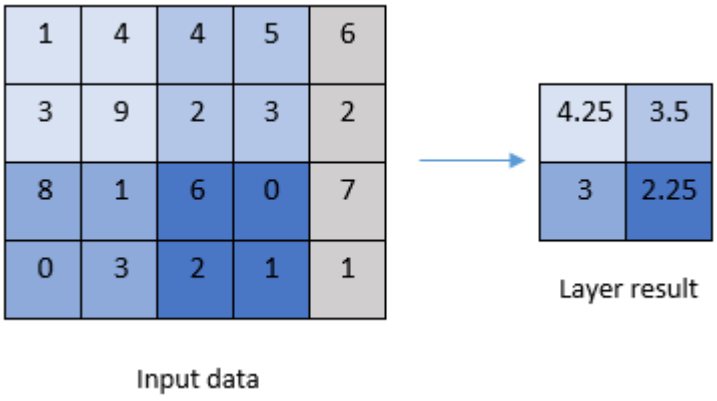


- CNN maken om de dataset te kunnen trainen

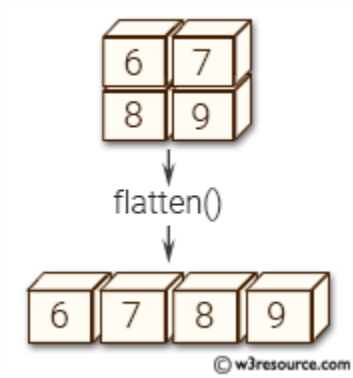
1. Transfer learning (mobileNet2)
   Transfer learning is een populier methode bij machine learning, het is hergebruiken van een pre-trained model voor een nieuw taak**.** Hoe meer de taken op elkaar lijken hoe meer de kennis van de ene bruikbaar zal zijn voor de andere. een model dat een goed onderscheid kan maken tussen honden en katten, kan snel leren om ook een onderscheid te maken tussen leeuwen en tijgers.
   Ik heb deze methode gekozen omdat het sneller is, er relatief weinig data en weinig computerkracht nodig is om het model de nieuwe taak te laten uitvoeren.



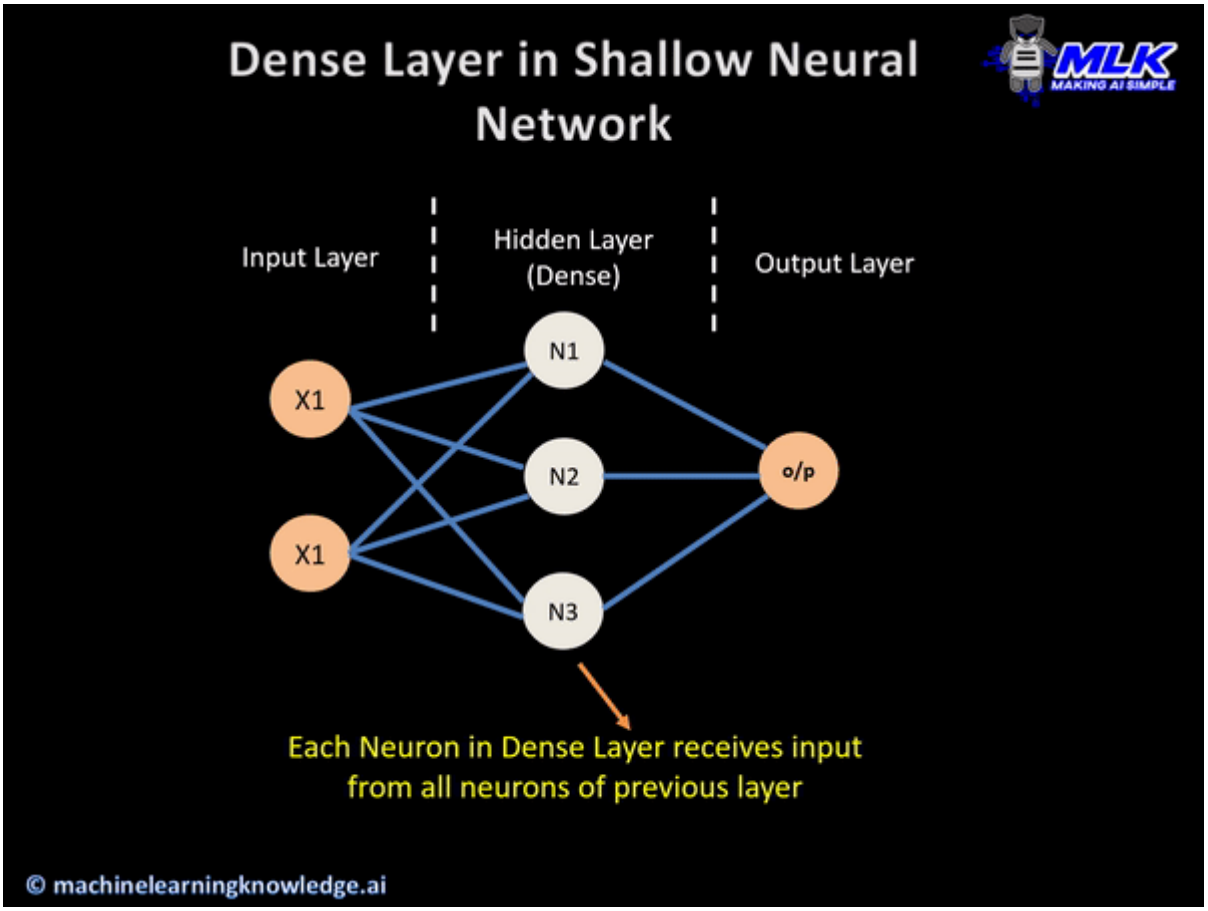
2. AveragePooling2D



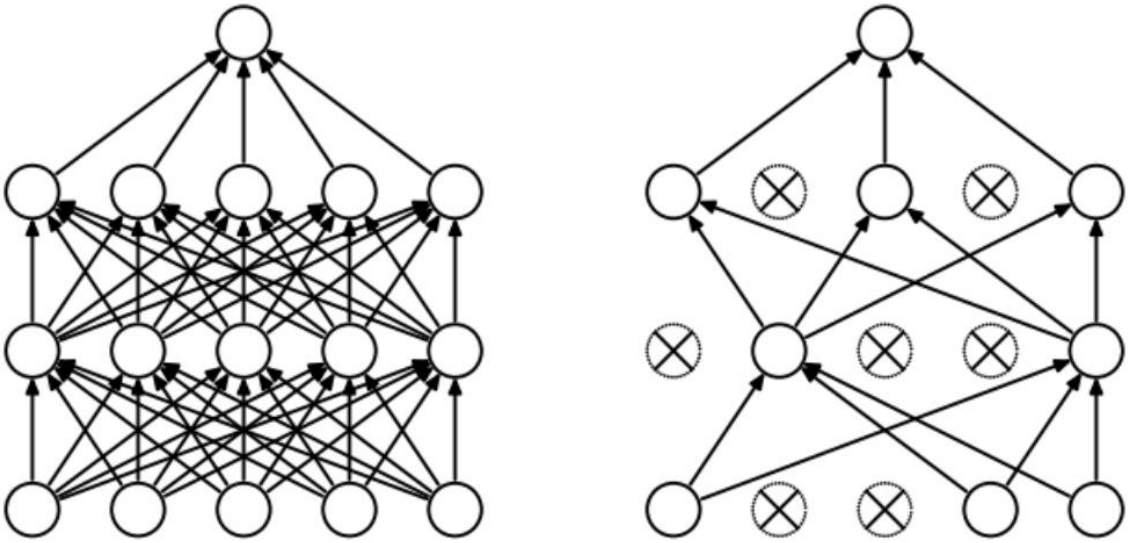
3. Flatten
   De functie flatten() wordt gebruikt om een kopie van een gegven array 1-dimensionaal te maken.



4. Dense layer is het meest gebruikte layer.



5. Dropout:



(a) Standard Neural Net (b) After applying dropout.

Experimenten

- ### Test code

Ik heb een test_face_mask functie gemaakt om mijn getraind model te kunnen testen. Ik heb mijn model op 33 foto's getest. 20 foto's met mondkapjes en 13 foto's zonder mondkapjes.