ONLY ONE GOOD FACE PER PICTURE - PROJECT

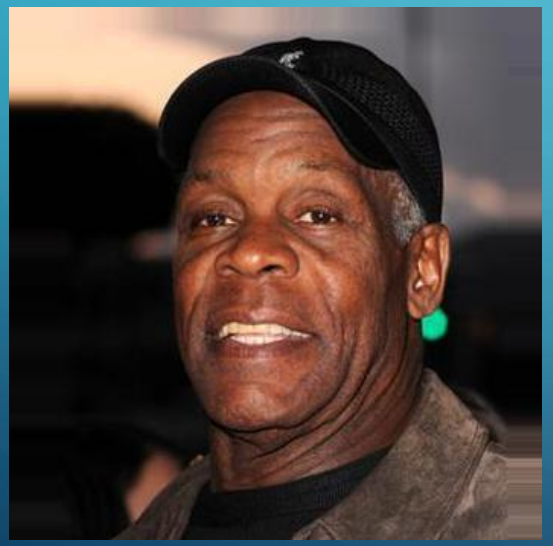
Maarit Salo

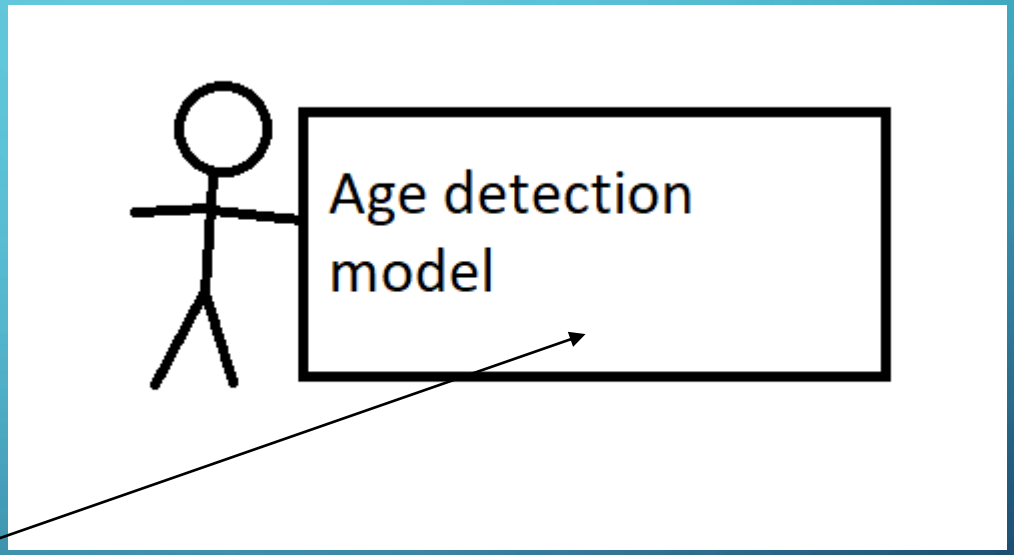
MAARIT.SALO@GMAIL.COM

HTTPS://GITHUB.COM/VIHERVIRVELI/PORTFOLIO

MY INSTRUCTOR HAS AN AGE DETECTION MODEL HE WANTS TO TRAIN…

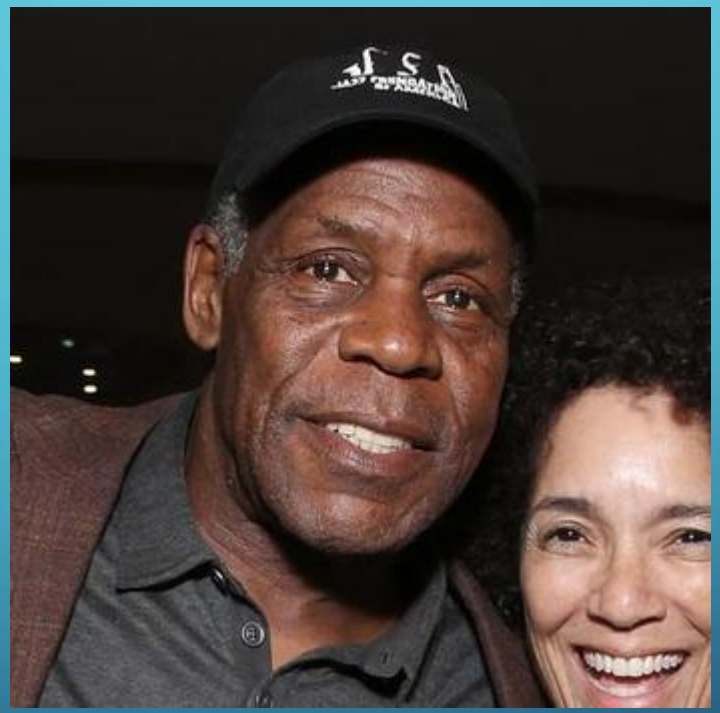
He wants pictures with only one face in them, like the one below





…BUT SOME OF THE PICTURES HE HAS LOOK LIKE THIS:

Age?



Age? (Which one?)

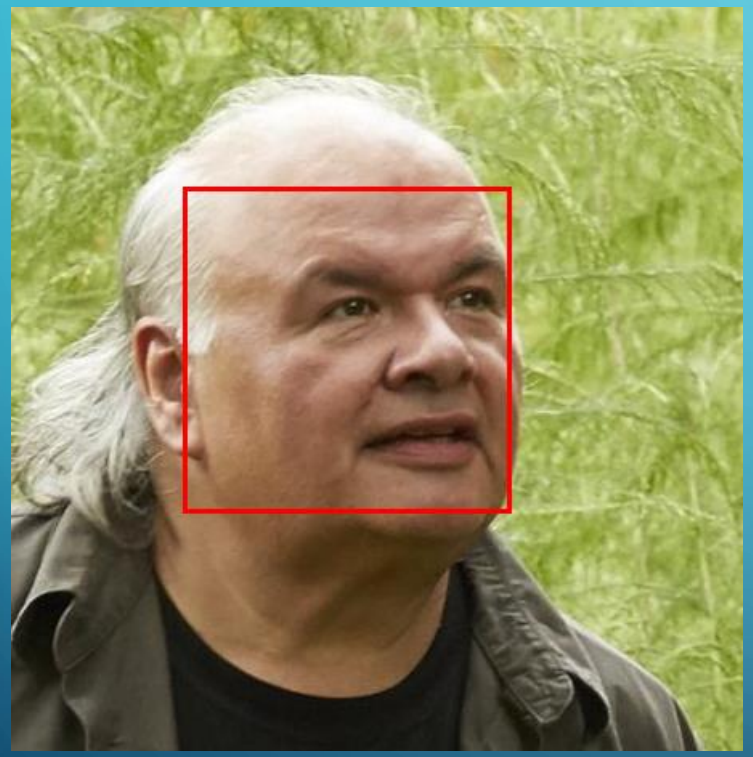
???

SO I WILL TRAIN A NEURAL NETWORK TO DISTINGUISH BETWEEN GOOD AND BAD PICTURES

Label: “Keep”

Label: “Delete”

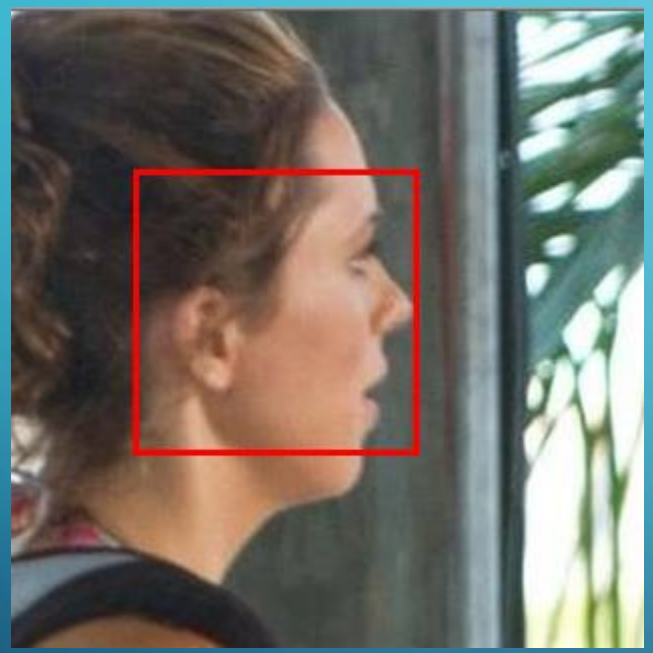
I USED FACE RECOGNITION TO ISOLATE PICTURES WITH ONLY ONE FACE IN THEM



They are labeled “keep”

THE PICTURES THAT HAD NO FACE AT ALL, OR TOO MANY FACES, WERE LABELED ”DELETE”

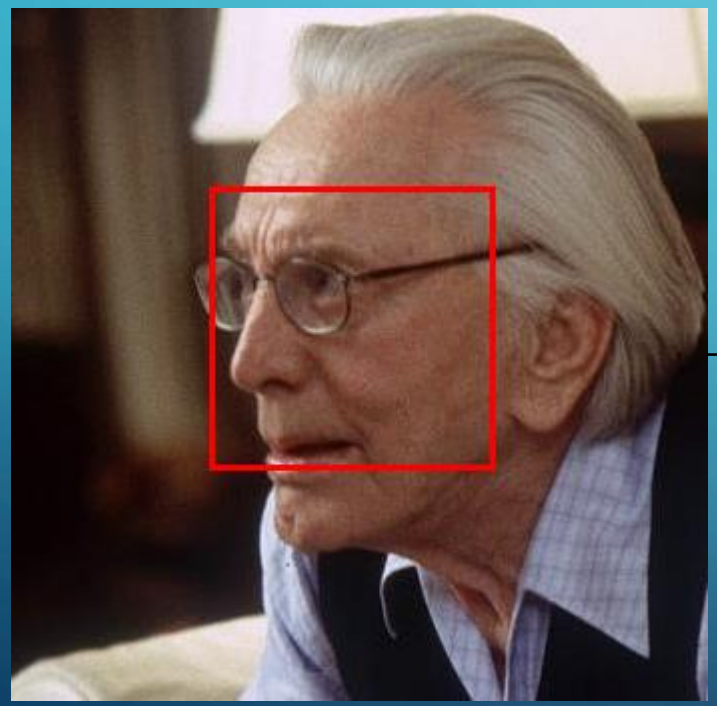
BUT FACE RECOGNITION ALONE WASN'T ENOUGH, BECAUSE IT ALSO GAVE ME PICTURES LIKE THIS AS "KEEP"

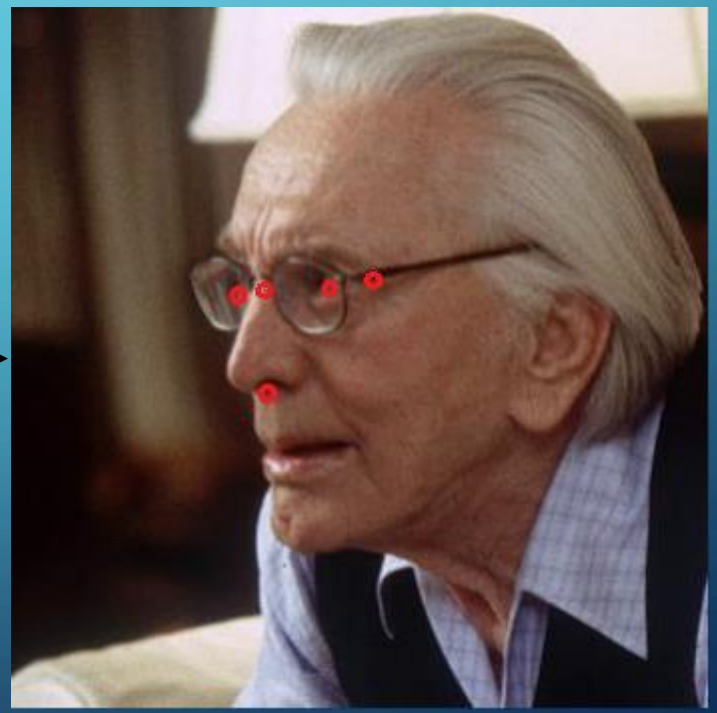


Sure, it's a face, but it's not very helpful for training an age recognition model

Pictures taken from the side are a problem for face_recognition

SO I ALSO RAN THE ”KEEP” LABELED PICTURES THROUGH FACE_LANDMARKS TO SEE IF IT FINDS A NOSE AND EYES IN THE PICTURE





BETTER RESULTS, THOUGH NOT PERFECT EITHER

Face_landmarks does not find a nose or eyes

Face_landmarks finds a nose, an eye and a ghost eye???

Given the time limit, I had to settle for these inconsistences.

NOW THAT OUR PICTURES ARE SORTED PER LABEL, TIME TO ASSEMBLE THE MODEL

These layers will help the model generalize better by "turning off" 10% of the neurons

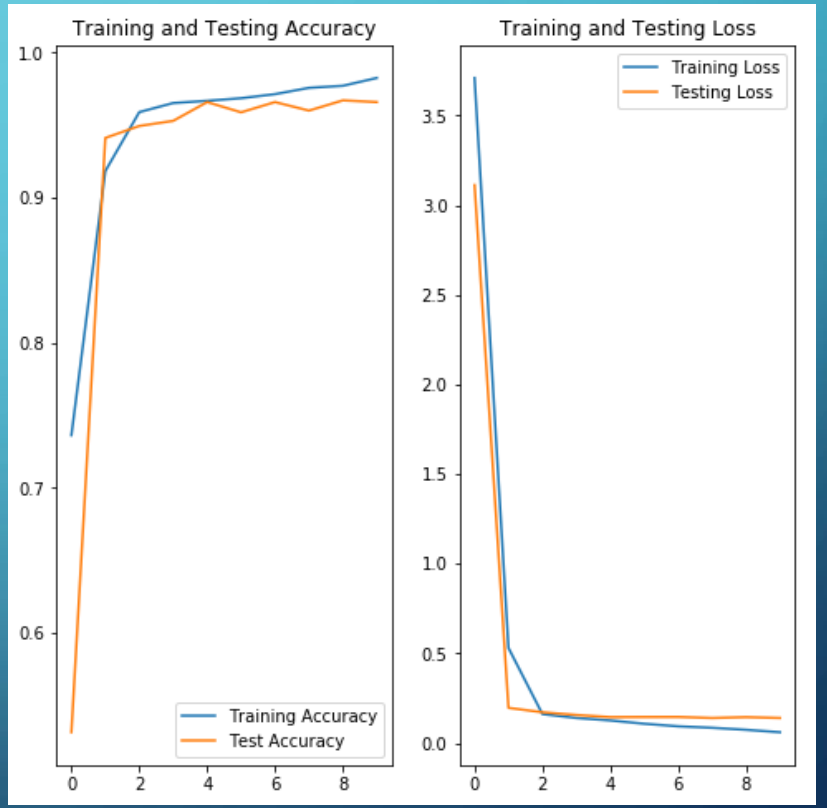
```
model = Sequential([
    Conv2D(16, 3, padding='same', activation='relu', input_shape=(IMG_HEIGHT, IMG_WIDTH ,3)),
    MaxPooling2D(),
    Dropout(0.1),
    Conv2D(32, 3, padding='same', activation='relu'),
    MaxPooling2D(),
    Conv2D(64, 3, padding='same', activation='relu'),
    MaxPooling2D(),
    Dropout(0.1),
    Flatten(),
    Dense(512, activation='relu'),
    Dense(1, activation='sigmoid')
])
```

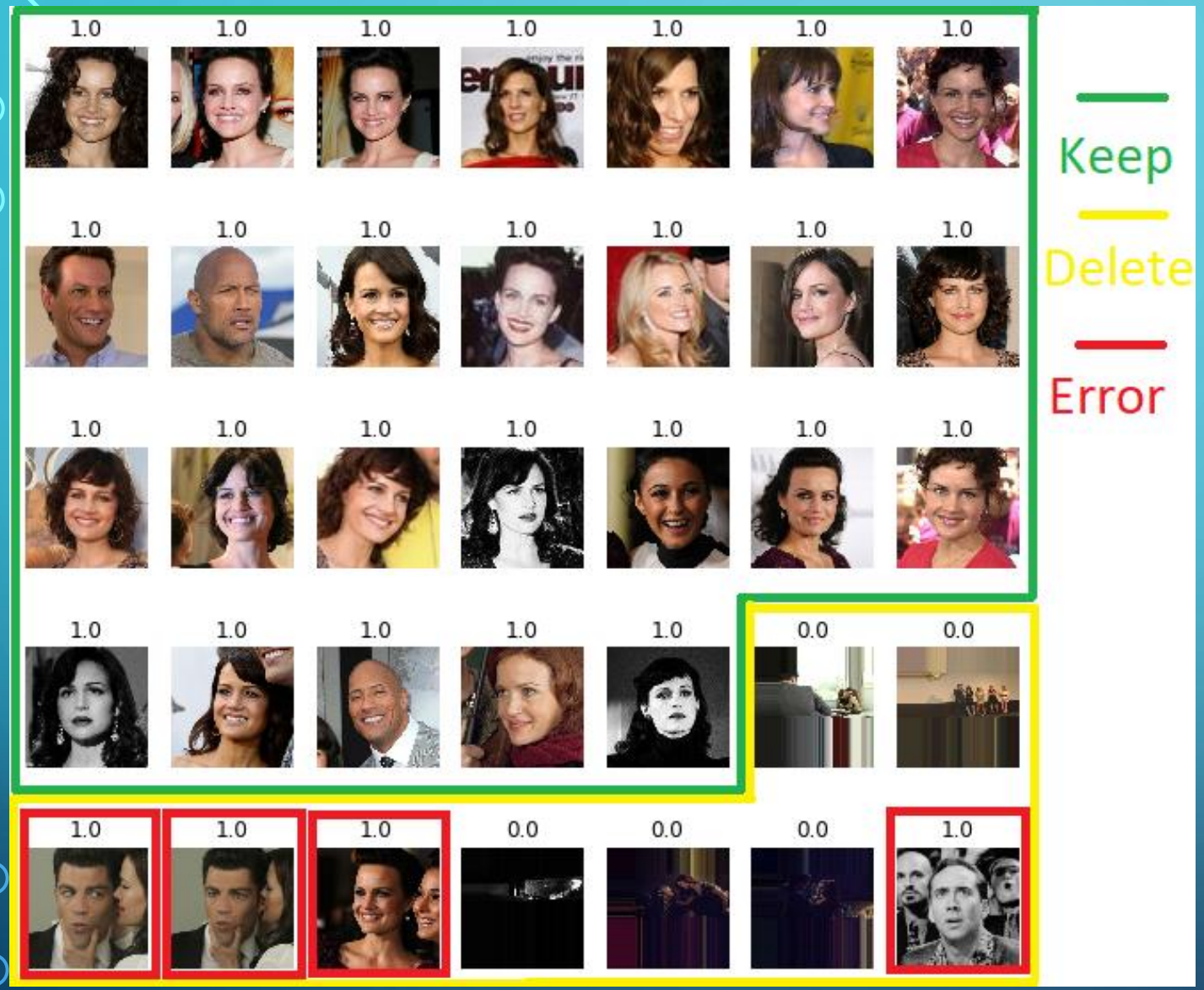
Generalizing means the model will work with new pictures, not just the ones I've given it

This single neuron will give 1 or 0 per picture, 1 meaning "keep" and 0 "delete"

GIVEN THE TIME LIMIT OF SIX WEEKS, I ACHIEVED THE BEST RESULTS POSSIBLE, AND THE MODEL WORKS

Accuracy goes up, loss goes down. Model.evaluate with new pictures gave an accuracy of 88.57% which is lower than here, but still pretty good





35 pictures, accuracy 88.57%

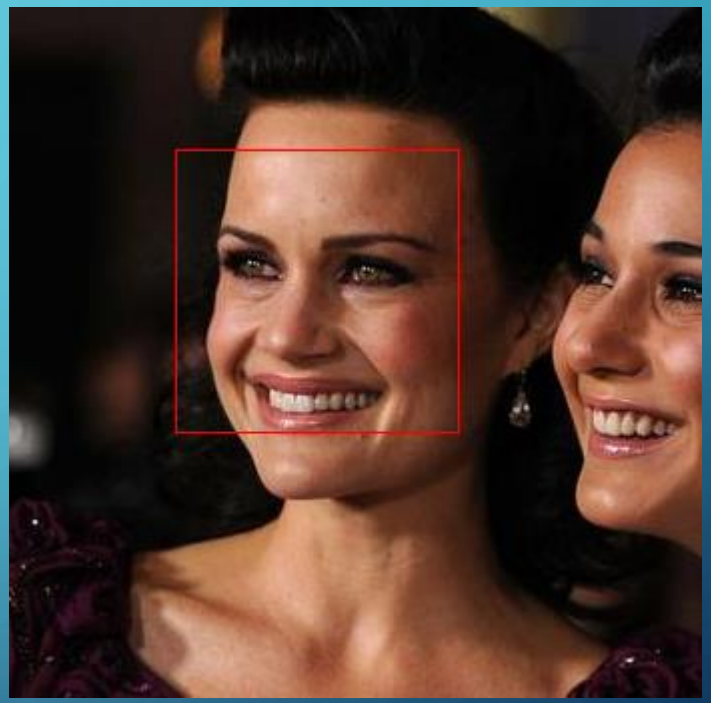
Correct = 35*0.8857 ~ 31

Error = 4 pictures

WAYS TO IMPROVE THE MODEL

- Label the pictures in a better way

This picture was labeled “delete” by face_locations and face_landmarks, despite it having only one face and face_landmarks. In contrast, the model correctly labeled it “keep”. So the model behaved correctly, but the pre-preparation checks didn’t.



TOOLS

- Tensorflow 2.0
  - Image classification
  - Neural networks
- Python face_recognition
  - Batch_face_locations
  - Face_landmarks
- Python PIL
- Python Matplotlib.pyplot
- Python Pandas
  - DataFrame

Maarit Salo
maarit.salo@gmail.com
https://github.com/vihervirveli/portfolio