# Detecting Face with YoloV3 and Facenet

**Group 2**
Sotheanith Sok
Grant Chen
Alex Pham

## Goal

The goal of this experiment is to perform facial recognition on a group of people utilizing existing and well-developed technologies such as Facenet and YoloV3. Given a set of images, the program will use the pre-trained YoloV3 model to extract faces from those images. Then, it will feed those extracted faces into the pre-trained Facenet model to map those faces onto 128 axes space. The classification is based on some threshold of the euclidean distance between points derived from anchors or most probable images belonging to a desired group and points derived from some images.

## Prerequisites

Due to the complex nature of getting Tensorflow-GPU and various other libraries running in the same environment, it is highly recommended to install packages using Anaconda. Thus, we have provided "requirements.yml" for windows and macOS for quick and easy installation of the required libraries

## Dependencies

- Environments:
  - Python3.7
  - Anaconda
- Packages:
  - Pillow,
  - ScikitLearn,
  - Numpy,
  - Tensorflow 2.1
- External Libraries:
  - Yolo
  - Facenet
- Pretrained Models/Data:
  - facenet.h5
  - YOLO_Face.h5
  - Yolo_anchors.txt
  - face_classes.txt
  - data.zip

## Setup

How to set up the environment:

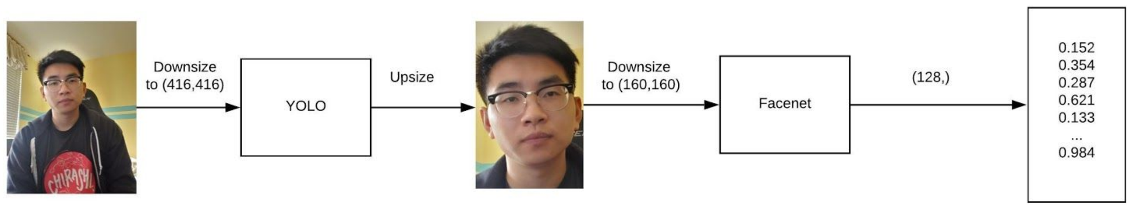
- Change the directory to the root of the projects
- Run "***conda env create -f ./winRequirements.yml***" for windows
  - It creates a new virtual environment called "**fy**" and installs all dependencies for windows
- Run "***conda env create -f ./macRequirements.yml***" for macOS

    - It creates a new virtual environment called “**fy**” and installs all dependencies for macOS
- Run “***conda activate fy***”
    - Activate the new environment

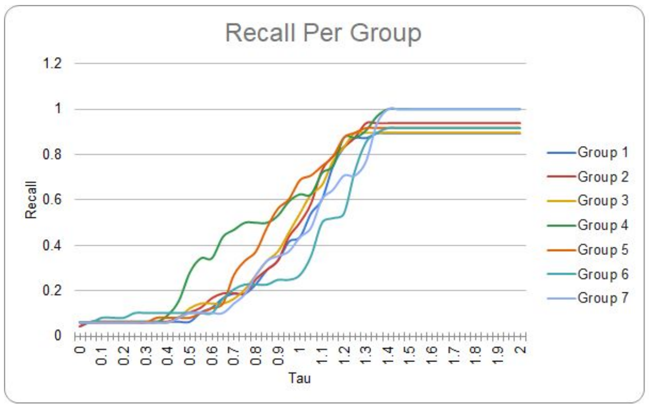
How to run the program:

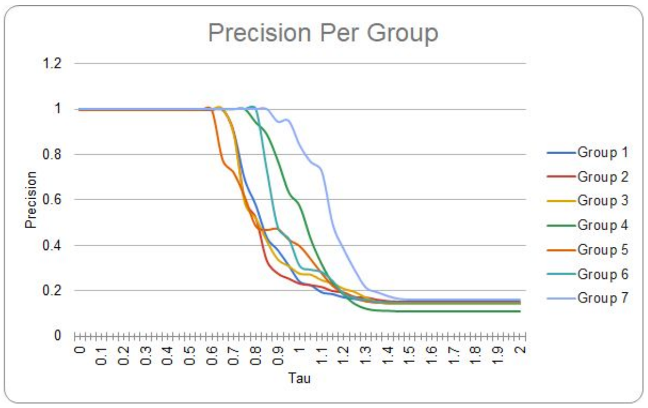
- Start the program by running “**python .\main.py**”
    - Optional flag:
        - “--tau [float]”: set the tau threshold to be used. Default: 1.0
        - “--show”: show a compilation image of all detected images
        - “--full”: run the algorithm on every group and a range of tau threshold
    - At the start of the program, it will download all necessary files from google drive. However, there are rare instances where the program fails to download files properly. Please manually download models/data from links above and put model files in “./models” and data.zip in “./data”

## Architecture

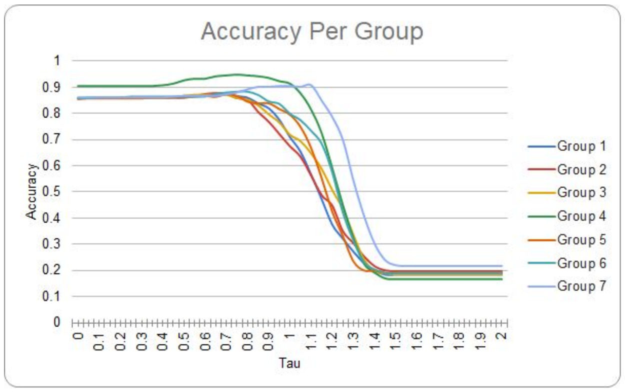


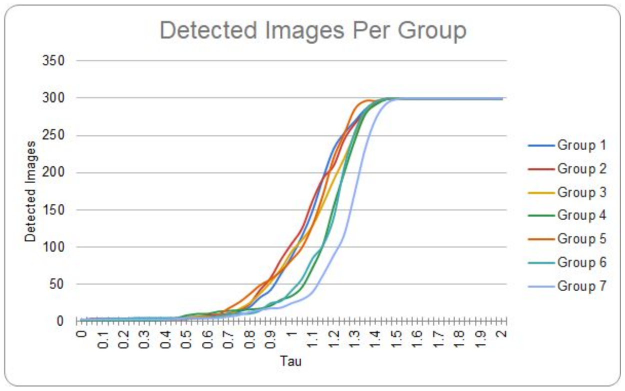
## Observation/Result



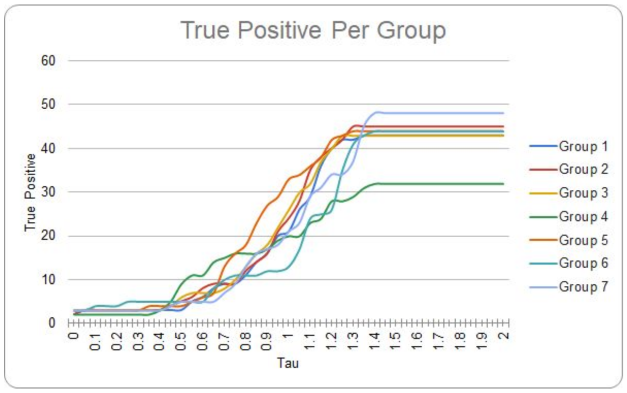


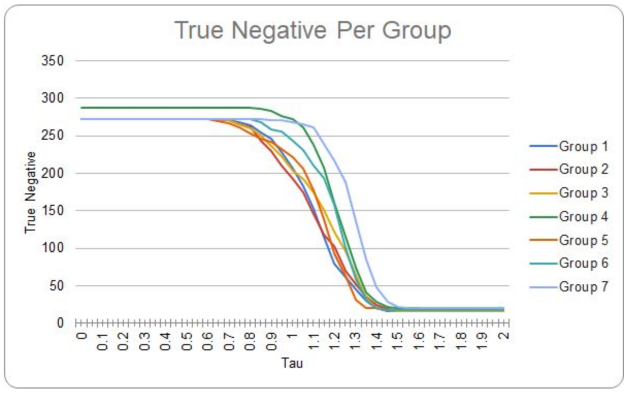
Based on the recall and precision graphs above, we can see that a threshold (Tau) of 1.0 is the best option. We want to maximize both and they are trending in opposite directions, therefore the median threshold is optimal. This is most likely because we did L2 normalization. A threshold of 1.5 results in a high amount of false positives which negatively affect the accuracy. Likewise, a threshold of 0.5 fails to classify the correct face.

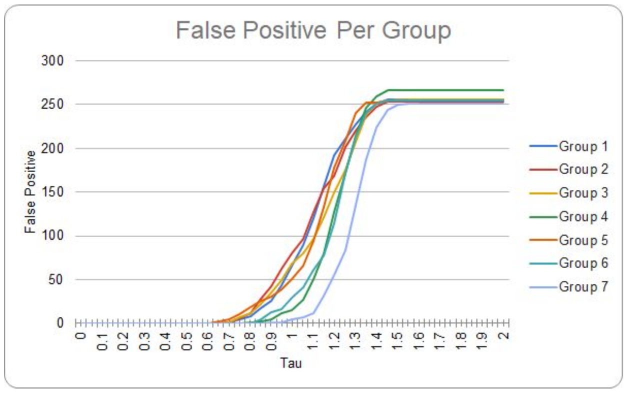


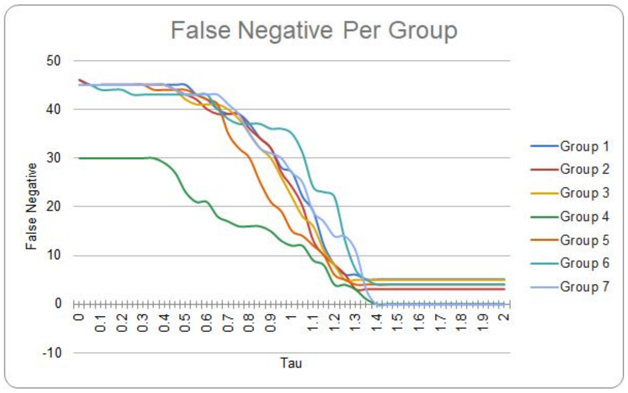


The first instinct some might have is to get as much data as possible to use. However, if we focused on maximizing the number of detected images, then we would have more false positives and the accuracy would plummet.









It is evident that at a Tau of 1.0 the false positives are relatively low while the true positives are at a reasonable point. If the threshold is high, then there are a high number of true positives and a low number of true negatives, but it would also result in a high false-positive rate. This is because more images are just being classified as positive identification. Not many are being rejected.

## Recommendations

Utilize precision and recall to choose the threshold since using these two metrics optimizes based on true positives. Using L2 normalization should allow a threshold of 1.0 to be the optimal choice.

## References

- Pre-trained YoloV3 by Thanh Nguyen: https://github.com/sthanhng/yoloface
- Pre-trained Facenet model by Hiroki Taniai: https://github.com/nyoki-mtl
- Florian Schroff, Dmitry Kalenichenko, James Philbin: FaceNet: A Unified Embedding for Face Recognition and Clustering 17 Jun 2015 (v3)
- Brownlee, Jason. "How to Develop a Face Recognition System Using FaceNet in Keras." *Machine Learning Mastery*, 21 Nov. 2019, machinelearningmastery.com/how-to-develop-a-face-recognition-system-using-facenet-in-keras-and-an-svm-classifier/.