

Fine tuning on VGG16 for Facial Recognition to Identify People of Interest

Jasper



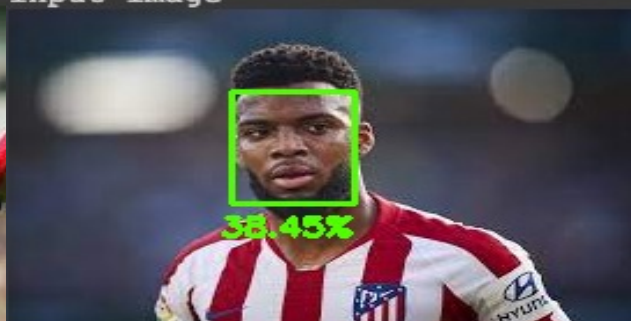
Found 1 faces!



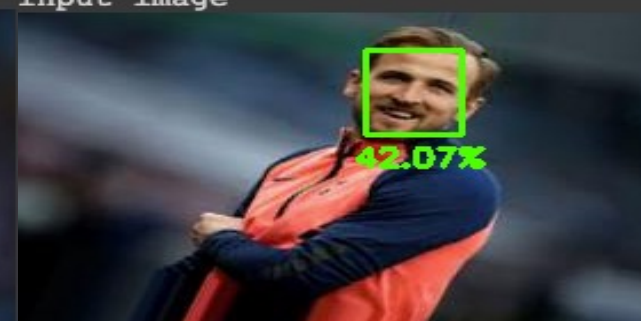
Found 2 faces!



input image



input image



input image



Problem Statement

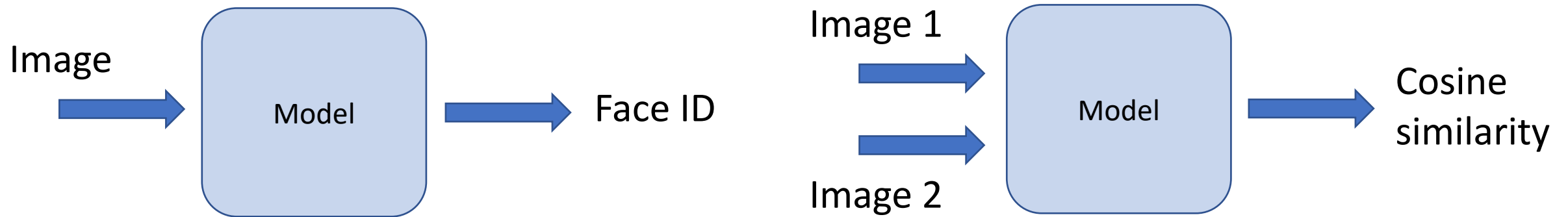
As Singapore is building towards a Smart Nation to improve the lives of citizen, the Smart City concept, with intelligent video analytics as one of its essential will become the basis of public security around Singapore.

This project aims to build a facial recognition model to identify people of interest which can be extended to aid law enforcement agencies such as ICA and the Police in identifying

- Missing people
- Criminals/Offenders
- Terrorists

Objective

1. Classification of the image of a face to the correct ID – metric used here will be the model accuracy
2. Similarity of faces in 2 different images – metric used here will be the cosine similarity

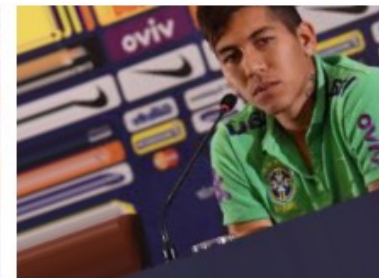


Dataset

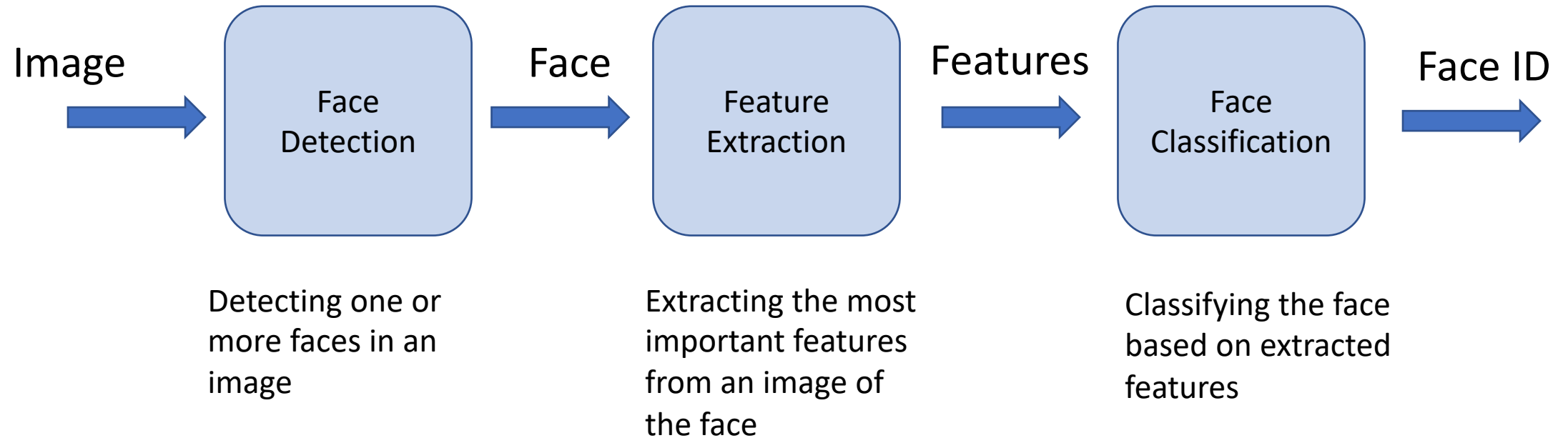
- Self-sourced of 20 images each for 100 soccer stars via Google
- Images of each soccer star were splitted into training, validation and test dataset with the ratio of 7:2:1 before image augmentation to prevent data leakage
- Image augmentation were done to expand the size of a training dataset by creating modified versions of images



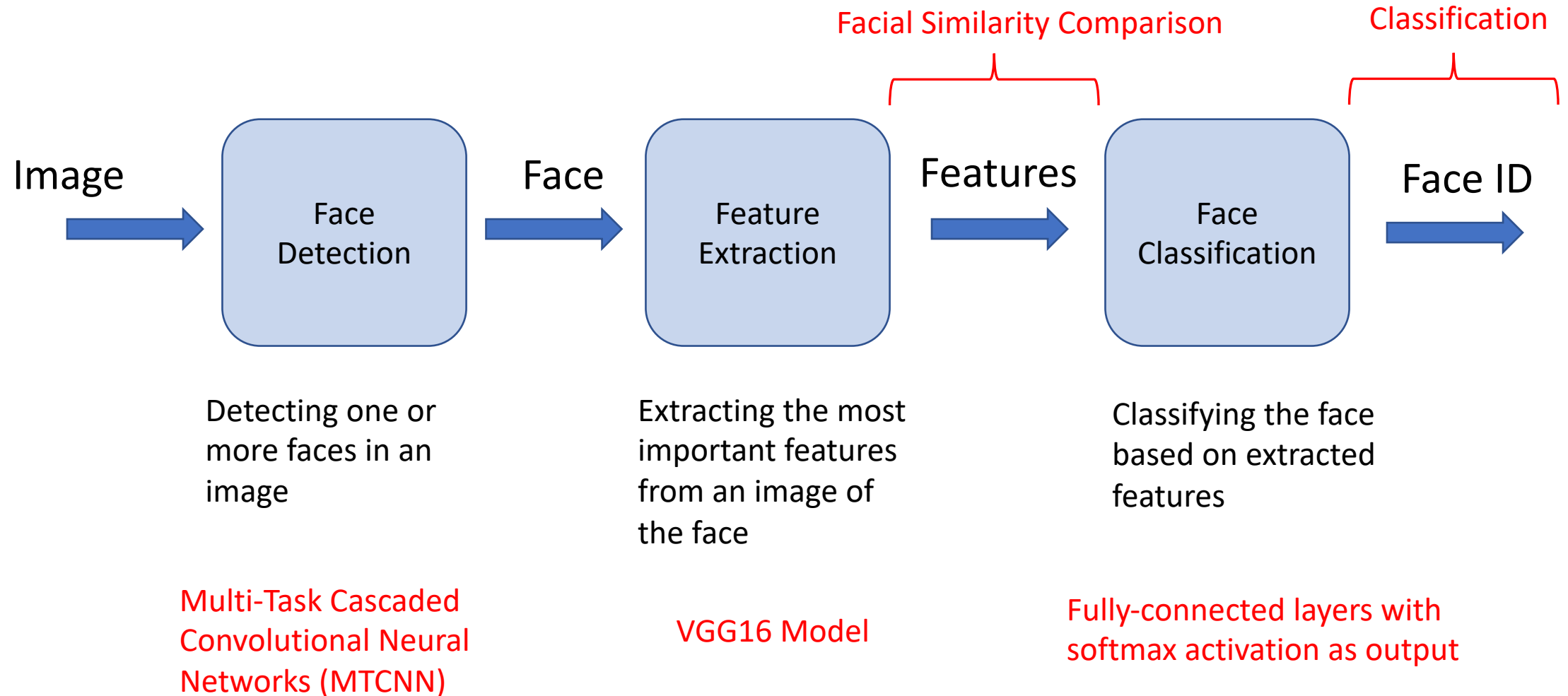
Image
Augmentation



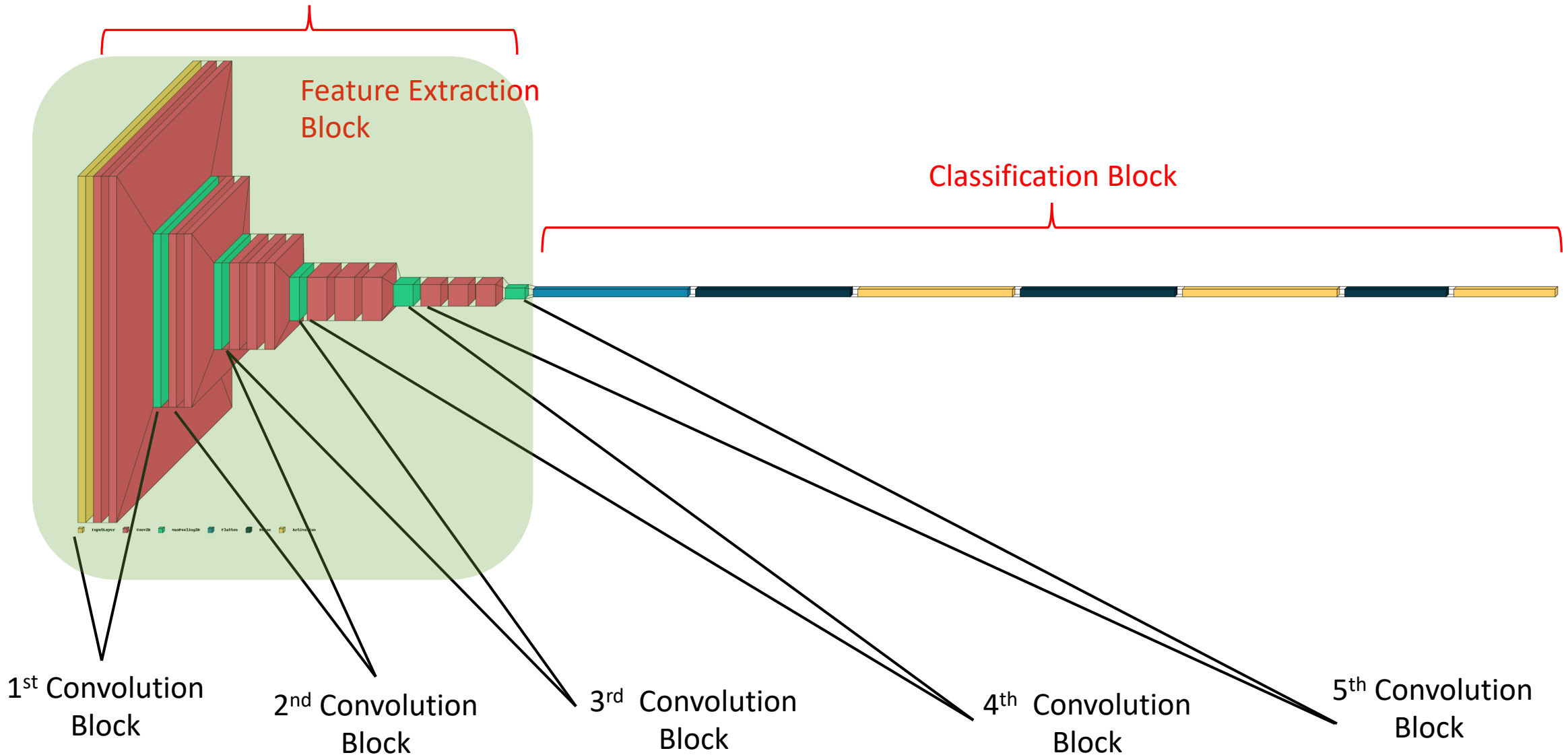
Methodology for Facial Recognition



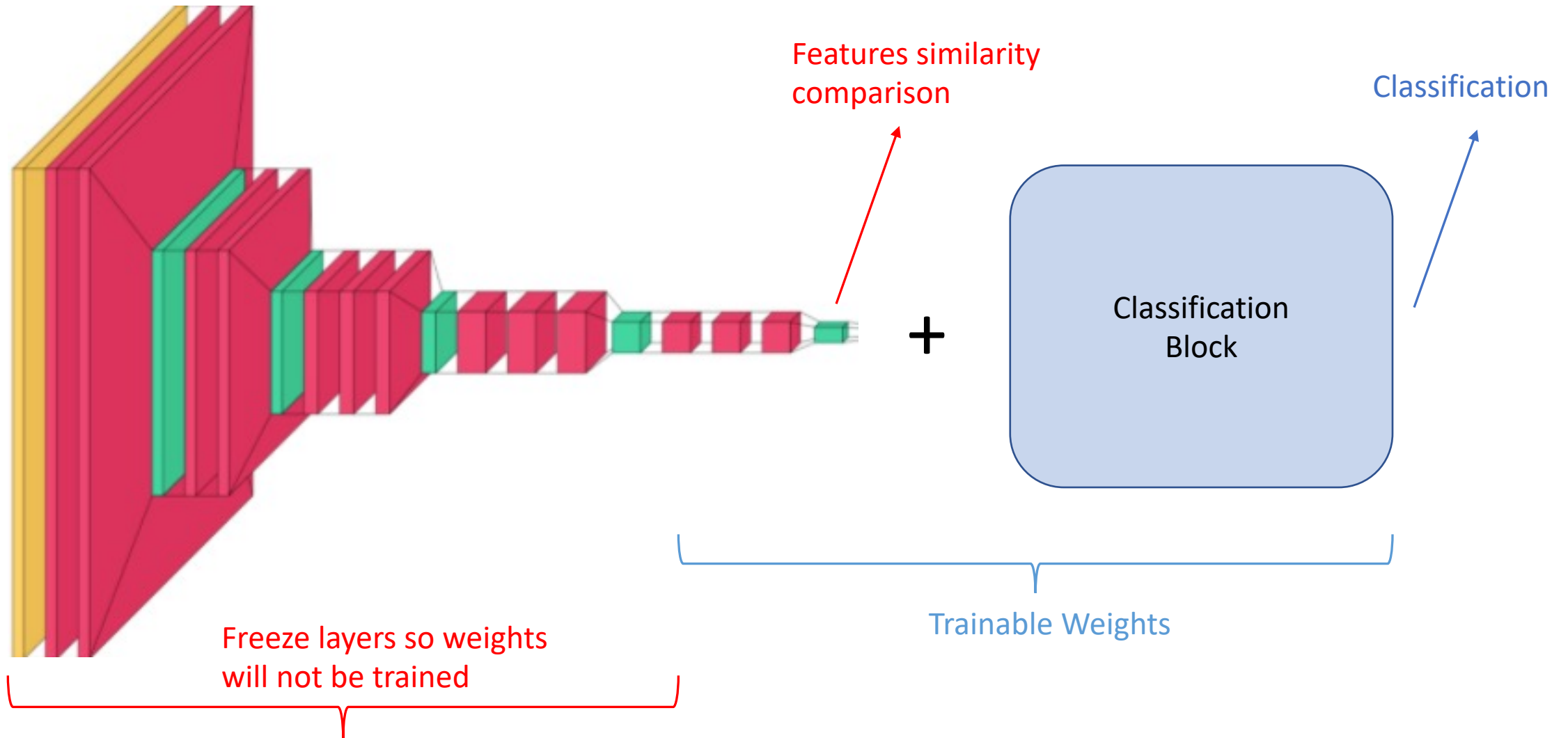
Methodology for Facial Recognition



Modeling Process – VGG16 Transfer Learning



Modeling Process – VGG16 Transfer Learning



Model Evaluation

Model	Hidden Layer Nodes	Dropout Layers Values	Training Accuracy	Validation Accuracy	Test Accuracy	Remarks
Base Model	[128, 128]	-	99%	73.2%	78.5%	Last layer of VGGFace trained
Base Model 2	[128, 128]	-	99%	64.5%	75.5%	Last 2 layers of VGGFace trained
Model 1	[128, 128]	[0.1, 0.1]	99%	75%	76.3%	Last layer of VGGFace trained
Model 2	[256, 256]	[0.3, 0.3]	97.8%	78.7%	81.5%	Last layer of VGGFace trained
Model 3	[512, 512]	[0.4, 0.4]	96.8%	77.2%	80.2%	Last layer of VGGFace trained
Model 4	[128, 128, 128]	[0.1, 0.1, 0.1]	98.8%	74.3%	78.2%	Last layer of VGGFace trained
Model 5	[256, 256, 256]	[0.3, 0.3, 0.3]	97.4%	75%	78.4%	Last layer of VGGFace trained
Model 6	[512, 512, 512]	[0.4, 0.4, 0.4]	96.7%	75.3%	77.1%	Last layer of VGGFace trained

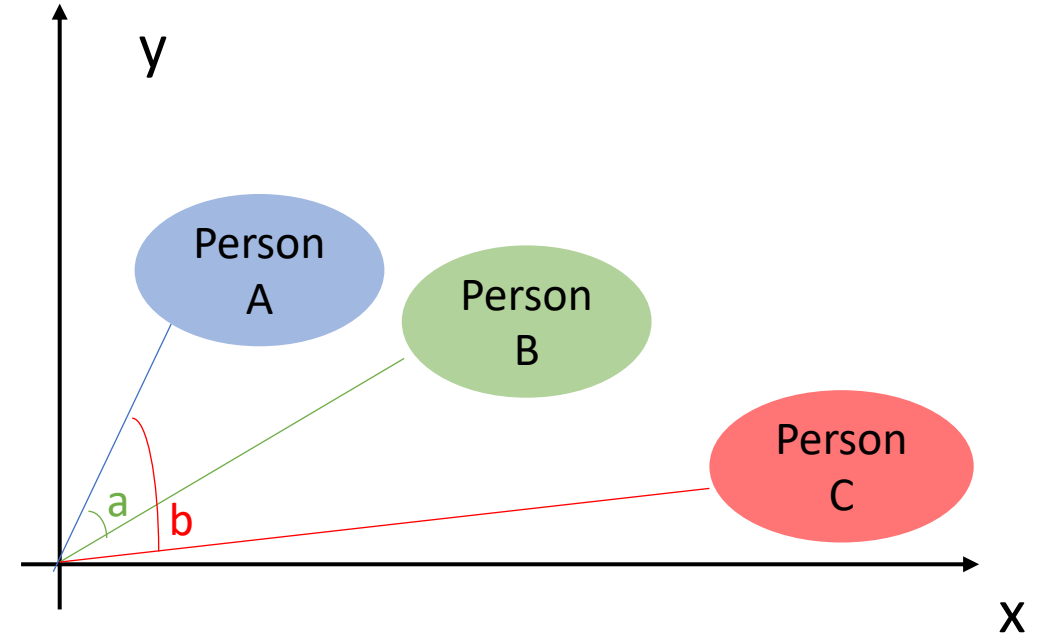
Model Evaluation

Model	Hidden Layer Nodes	Dropout Layers Values	Training Accuracy	Validation Accuracy	Test Accuracy	Remarks
Base Model	[128, 128]	-	99%	73.2%	78.5%	Last layer of VGGFace trained
Base Model 2	[128, 128]	-	99%	64.5%	75.5%	Last 2 layers of VGGFace trained
Model 1	[128, 128]	[0.1, 0.1]	99%	75%	76.3%	Last layer of VGGFace trained
Model 2	[256, 256]	[0.3, 0.3]	97.8%	78.7%	81.5%	Last layer of VGGFace trained
Model 3	[512, 512]	[0.4, 0.4]	96.8%	77.2%	80.2%	Last layer of VGGFace trained
Model 4	[128, 128, 128]	[0.1, 0.1, 0.1]	98.8%	74.3%	78.2%	Last layer of VGGFace trained
Model 5	[256, 256, 256]	[0.3, 0.3, 0.3]	97.4%	75%	78.4%	Last layer of VGGFace trained
Model 6	[512, 512, 512]	[0.4, 0.4, 0.4]	96.7%	75.3%	77.1%	Last layer of VGGFace trained

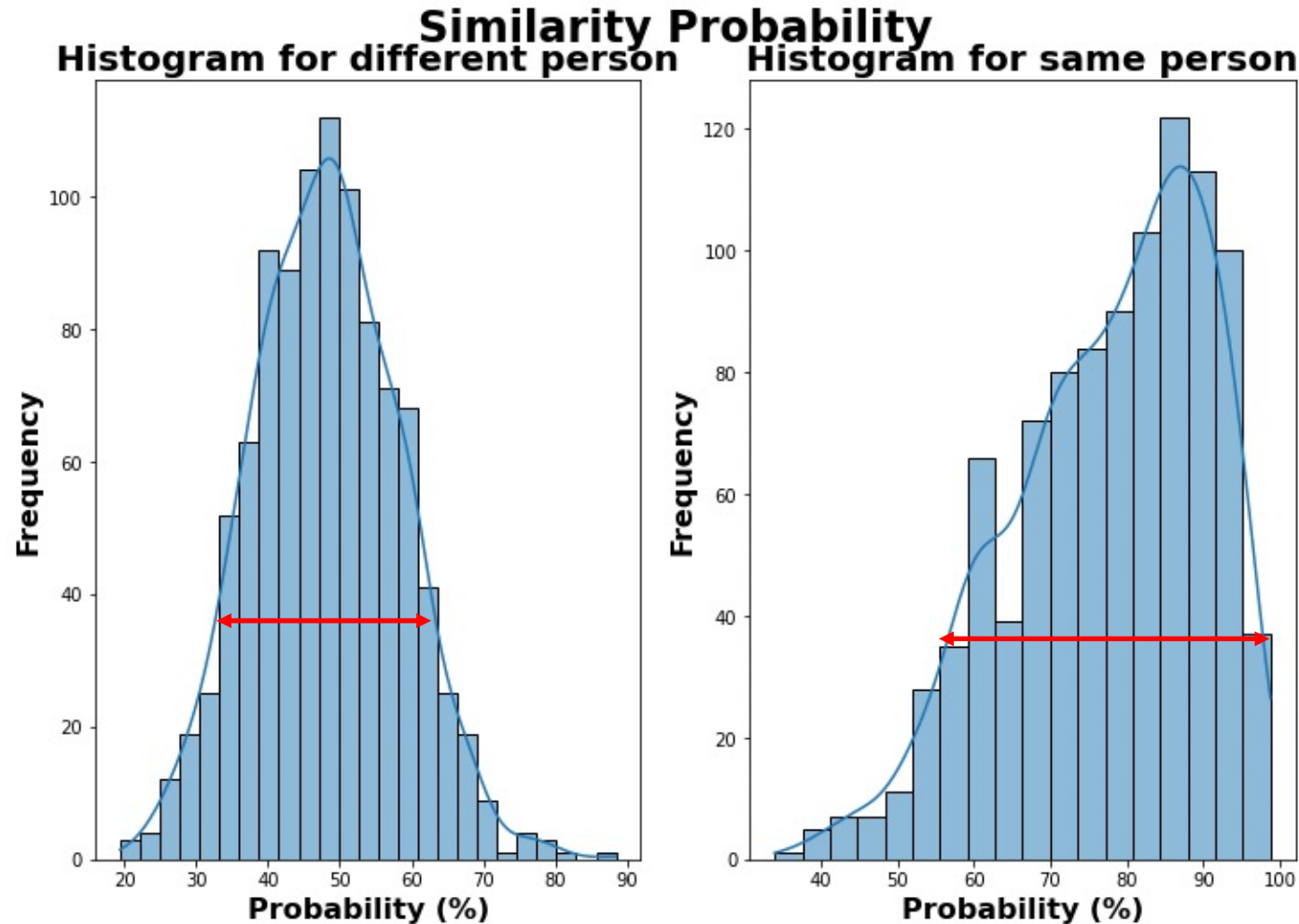
MODEL 5 Chosen

Facial Features Similarity

- Cosine similarity – a metric that measure the similarity between 2 vectors in high-dimensional space
- Represent the probability of how similar 2 images are



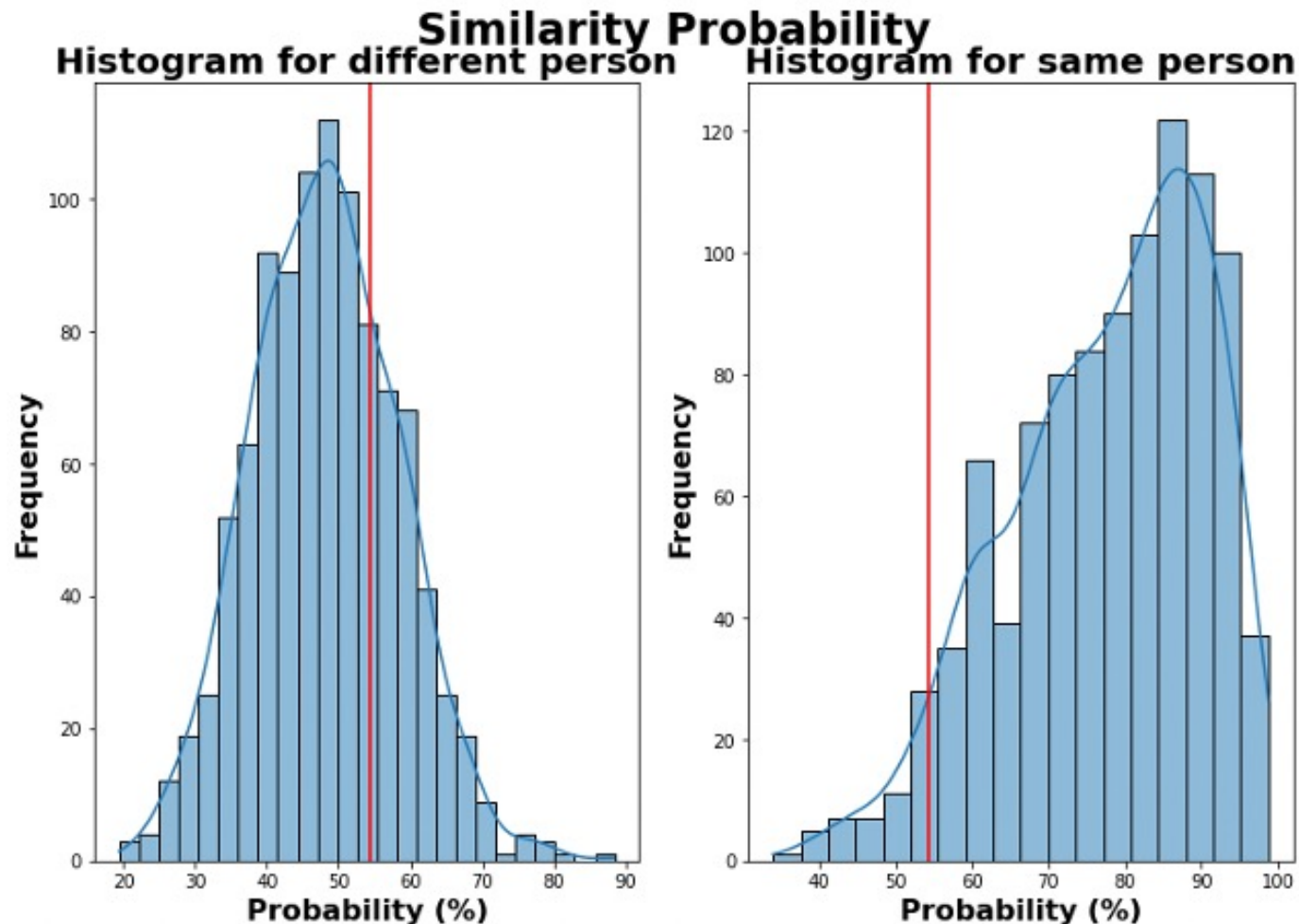
Facial Features Similarity - Model 5



Cosine similarity for a random samples of 1000 pair images

- I. 2 images of same person
- II. 2 images of different person

Facial Features Similarity - Model 5



- Threshold set at 54.4%
- 95% of the pair-images to correctly identify the images of the same person

Demo - Classification



Demo – Facial Similarity



Input Image

Output Video

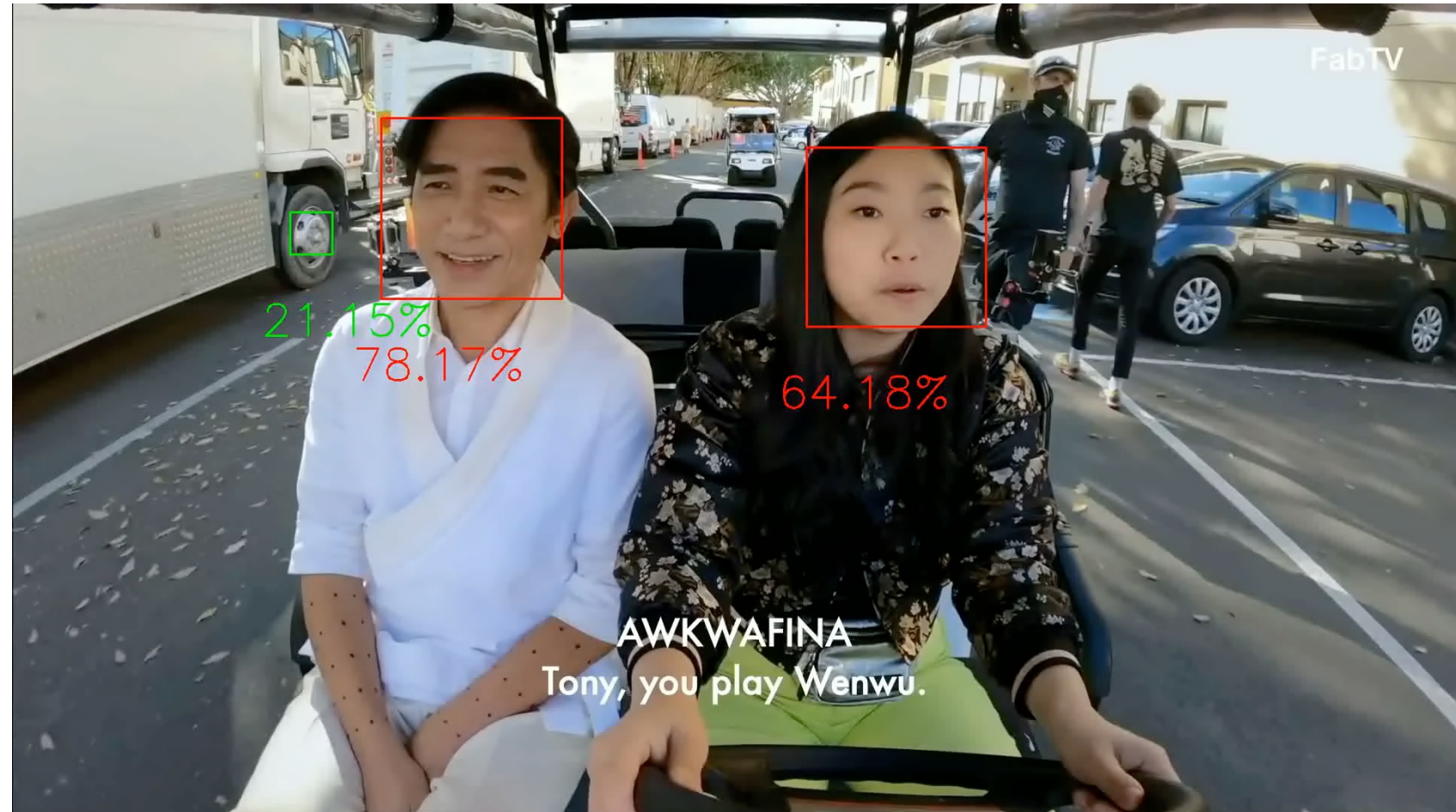


Demo – Facial Similarity

Output Video



Input Image



Conclusion and Recommendation

- Model 5 chosen with considerably good accuracy of 78.4%, together with good performance for cosine similarity for images of same person and images of different person
- Model is able to identify people in the video
- Given an input image, the model is able to output the probability of the faces in the video that is similar to the input
- Improve on the face detector to be able to detect just the face only
- Implement Triplet loss as the loss function to improve the accuracy
- Repeat more experiments of the 1000 random samples to be more confident of the threshold being chosen

THANK YOU !

Q & A?