Computer Vision Final Project

FACIAL EMOTION RECOGNITION ON MOBILE DEVICES

Pedro Moreira, Giuseppe Cianci, Leonida Lumburovska

June, 2024

MOTIVATION

- current chatbots do not consider emotions from the beginning of a communication flow
- integration of vision and interaction
- mobile app to enhance accessibility and UX

PROJECT IDEA

- face recognition
- consideration of facial expressions of user by using real-time emotion detection
- personalized interactions

MODELS

Face recognition

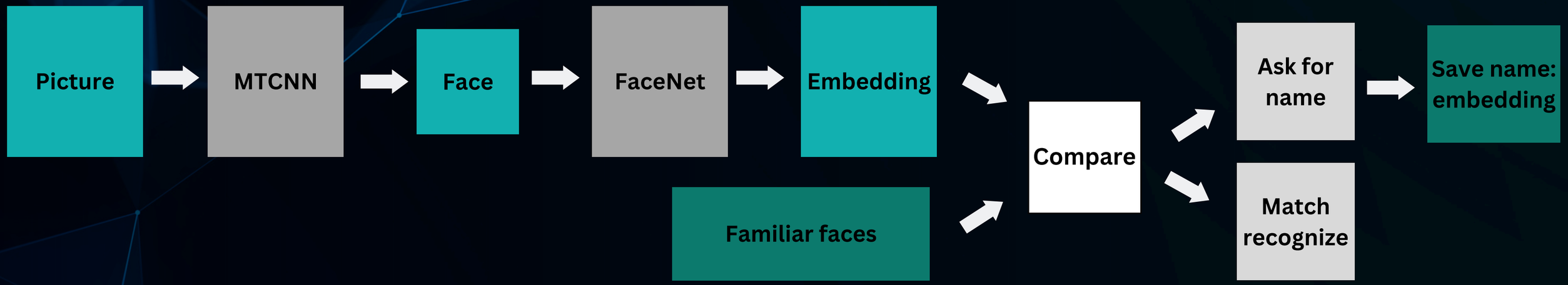
MTCNN
FaceNet

Emotion detection

Custom made EfficientNet-like network trained on FER-2013 and

FACE RECOGNITION - MTCNN

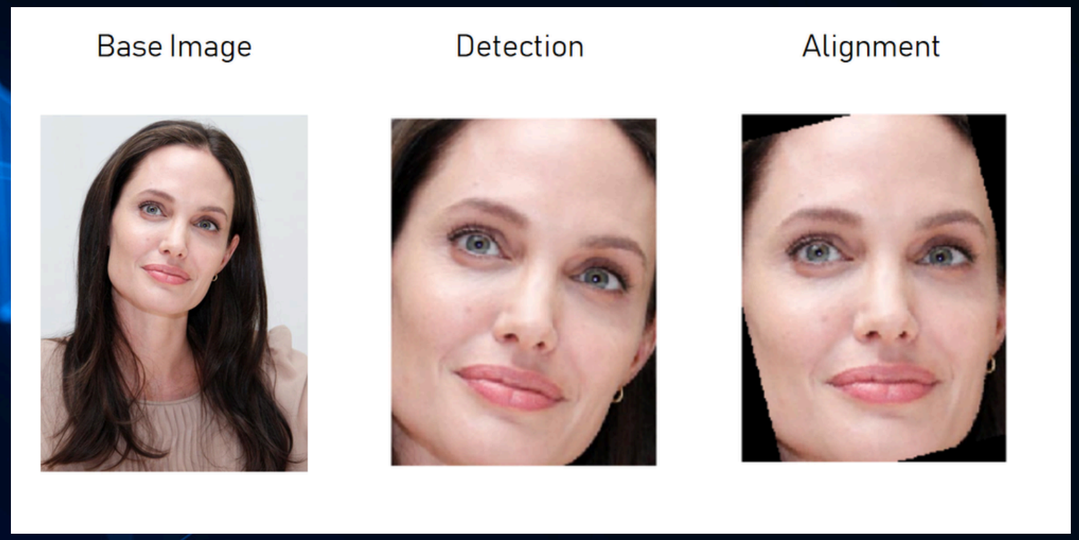
FaceNet for face embeddings
Multi-Task Cascaded Convolutional Neural Network for face detection

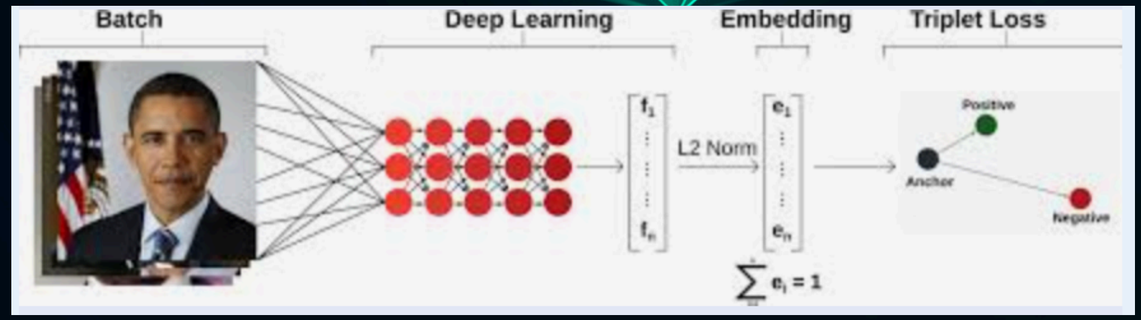


FACE RECOGNITION - MODEL VARIANTS

- Multiple Facenet TensorFlow Lite models depending on
  - dimensionality of the output embedding
  - precision of the weights and activations
  - pruning (eliminating parameters with low impact)

FACE RECOGNITION APPLIED

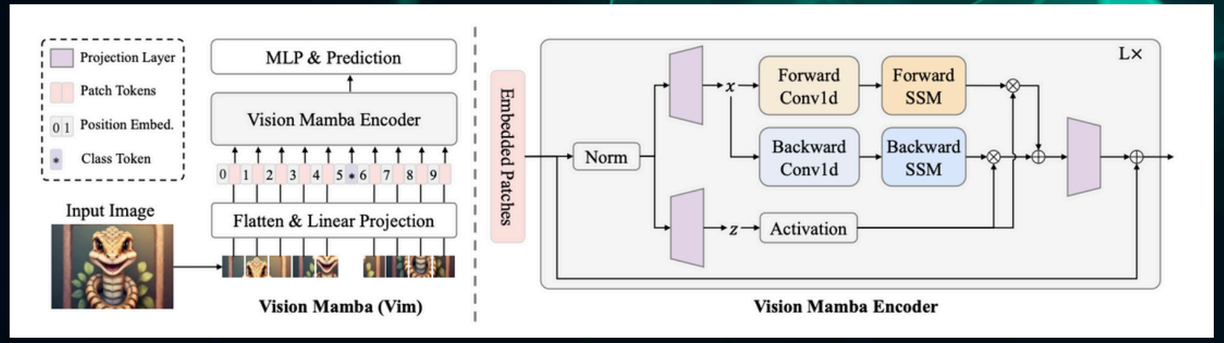


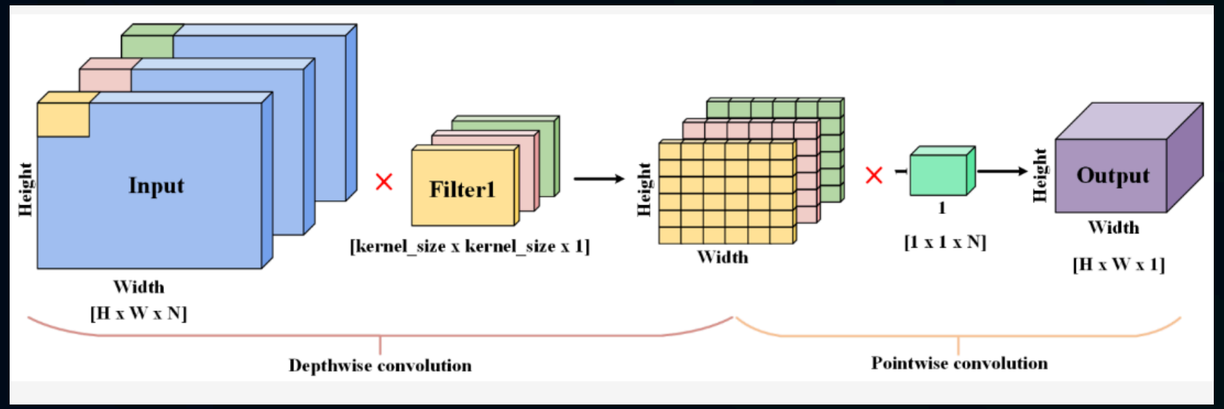


EMOTION DETECTION -ATTEMPTS

SSM (Mamba Vision)
(more favoured for unlabeled data, heavily relies on both the quality and quantity of data)

EfficientNet
(suitable for real-time applications, optimized scaling strategy)





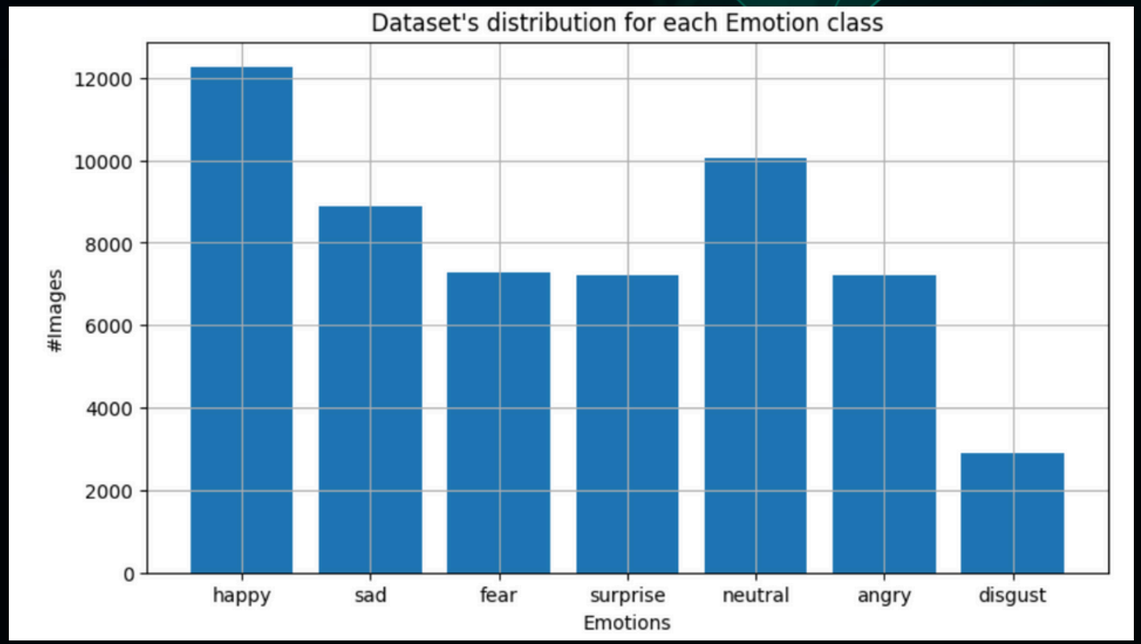
EMOTION DETECTION

EfficientNet- like architecture

- Compound Scaling
- Efficient Building Blocks
- Swish Activation
- Mobile Inverted Bottleneck Convolution
- Global Average Pooling and Dropout

DATA

- 48x48 pixel grayscale images of faces
- Labelled data: (0=Happy, 1=Sad, 2=Fear, 3=Surprise, 4=Neutral,
- 5=Angry, 6=Disgust)
- Train-test split ~ 90-10
- 55775 images



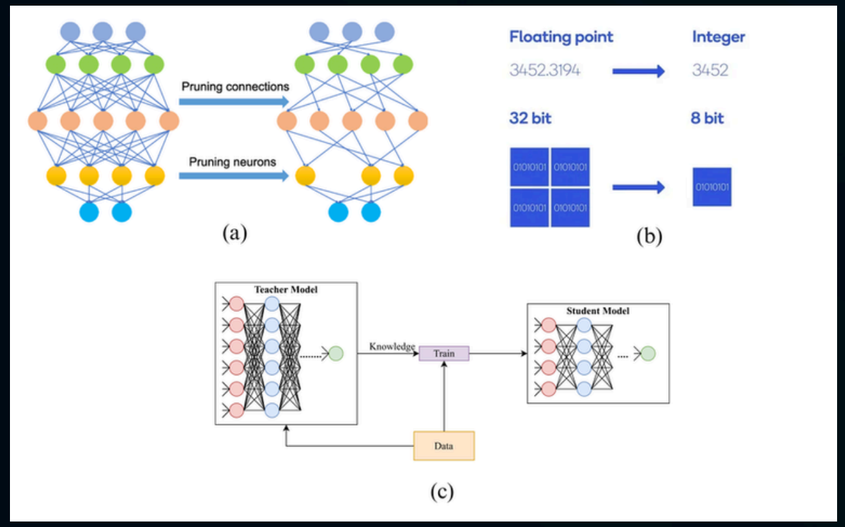
ARCHITECTURE

- Initial Conv2D layer
- 7 MBConv blocks
  - Depthwise Seperabale Convolutions
  - SE blocks
- Final Layers
  - Global Average Pooling
  - Dropout
  - Dense layer using softmax activation

PRUNNING & QUANTIZATION

- **Transforming to TFLite:** Converts the model to TensorFlow Lite format without pruning or quantization by default.
- **Pruning**: Removes unnecessary weights from the model, typically done during training. From 32bit float to 16 bit
- **Quantization**: Reduces the precision of weights and/or activations, done during or after training.

TensorFlow Lite

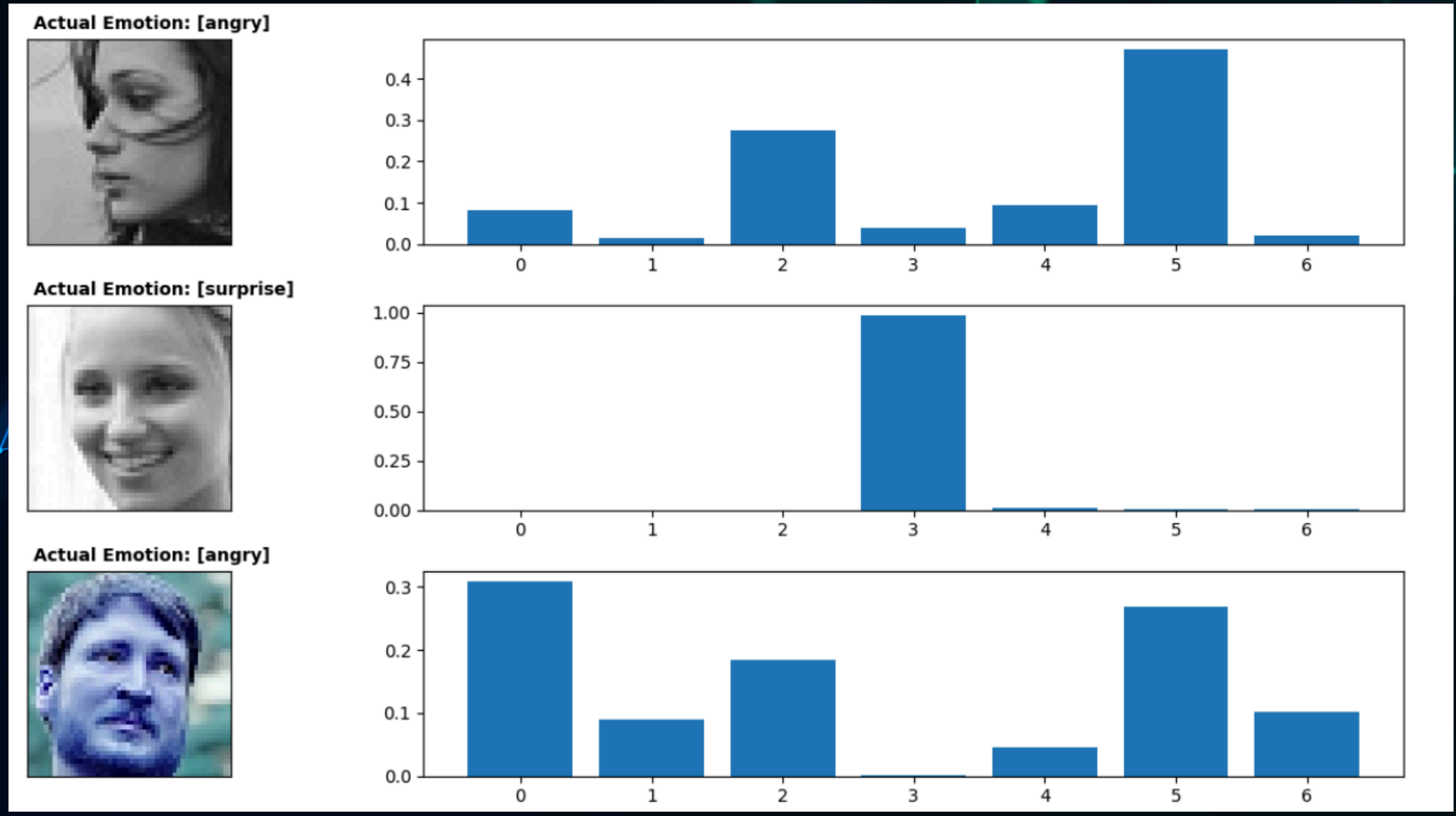


RESULTS

- 100 epochs
- 0.001 learning rate
- Batch Size: 64
- Validation accuracy: 57.15%
- Loss (categorical crossentropy) : 1.102

RESULTS

0=Happy
1=Sad
2=Fear
3=Surprise
4=Neutral
5=Angry
6=Disgust



EXTRA - OBJECT RECOGNITION

YOLO finetuned on Object Images in Hand

From multiple videos with objects we extracted every frame and used those images to finetune the model