**CS5131 Project Report**

**Title: Facial Recognition Model using One-shot Learning as an alternative to boarding fingerprint scanners**  
  
**By: Choy Aik Lok & Dominic Cheong**

**Objective**This project intends to provide an alternative to the fingerprint scanners at our boarding school using facial recognition. The project only intends to cover the backend, and there will be no prototype. The hardware used for demonstration will be laptop webcam.

**Development Processes and Tools Use**

In order to run the notebook, just download the given libraries at the first cell and run the whole notebook.

Libraries used: OpenCV, Tensorflow, Tensorflow Keras, Matplotlib

Development Processes:

* Data Collection — Using the OpenCV library and the publicly available Cascade Classifier, images captured by the device camera can be processed by an algorithm to detect faces, draw a bounding box around them and store them in an appropriately named file.
* Model Architecture — The Tensorflow library allows for complex models to be built layer by layer to fulfil specific purposes. In this notebook, a simple transfer learning model was built from the existing mobilenetV2 model publicly available on the Tensorflow Keras website. In addition to that, a more complex triplet model was also built using guidelines from an online Kaggle notebook.
* Model Training — Both models are trained using Tensorflow’s .fit function. For the triplet model, a custom triplet loss function had to be built, also referencing the Kaggle notebook.

**Application of concepts and techniques**

This project has made use of the concepts of CNN, transfer learning, metrics to compare models (accuracy and precision), triplet loss, as well as python libraries such as OpenCV.

**Results and Findings**

**Screenshots and Diagrams**

Transfer Learning Model

**A screenshot of a computer program

Description automatically generated**

Triplet Model and Triplet Loss

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

Triplet loss


Model Diagram for the actual model is too large to be displayed here. You can find it under the Kaggle notebook reference.

**Limitations**

Tensorflow backwards compatibility issues caused issues when running code from the Kaggle notebook and loading the vggface model. Many of the fixes aren’t 1-to-1 representations of the original code.

Our models assume the person does not change, for both the testing and data collection process. If the person wears different facial accessories, our model will have trouble recognizing them.

**Conclusion and Recommendations**

Deep learning architecture is very useful in performing classification tasks and can be used in a variety of situations, as demonstrated by this project. However, the step-up from deep learning to one-shot learning is a difficult process.

In the future, it would be helpful to generate the embeddings and analyze the result to trace the source of error. Another useful but arduous task would be to study what each layer did to find the layer where errors originate from.

**Individual Reflections**

Dominic:

This project has been enlightening and has opened my eyes to the usefulness of AI and Machine Learning. In this project, I learnt how to use OpenCV, various TensorFlow libraries and the triplet model – triplet loss architecture. Despite the challenges faced in tracing vague error messages in the attempt to train the model, I was able to better understand the inner workings of machine learning models.

**References**

[1] <https://www.youtube.com/watch?v=lH01BgsIPuE>

[2] <https://www.kaggle.com/code/amankumarmallik/one-shot-learning-for-face-verification>

**Work Distribution Matrix**

|  |  |  |
| --- | --- | --- |
| **Work Description** | **Aik Lok** | **Dominic** |
| Ideation & Proposal | ✔ | ✔ |
| Data Collection & Method | ✔ |  |
| Model Type 1: MobileNet Transfer Learning CNN | ✔ |  |
| Model Type 2: Advanced CNN Model utilising Triplet Loss Function | ✔ | ✔ |
| Model Type 2: Data Generation for Triplet Loss Function |  | ✔ |
| Evaluation of Models | ✔ |  |
| Video | ✔ |  |
| Report | ✔ | ✔ |
| Finding References | ✔ | ✔ |