WQD 7004 - PROGRAMMING FOR DATA SCIENCE

SEMESTER II - SESSION 2018 / 2019

GROUP PROJECT TOPIC

TRAINING A CONVOLUTIONAL NEURAL NETWORK TO DO FACIAL RECOGNITION USING BIG DATA

GROUP MEMBERS

CHEAH JUN YITT	WQD180107
TAN YIN YEN	WQD180108
CHOO JIAN WEI	WQD180124

LECTURER

DR HAMID TAHAEI

SUBMITTED BY

26th MAY 2019

Contents

-

Introduction

Facial Recognition

Facial recognition system is a technology used to identify or verify a person from a digital image of the person. The application of facial recognition system is immense. It has been used in various systems such as access control in security systems, commercial identification and marketing tool, video surveillance, indexing of images, and various social media platforms related functionalities.

On November 2017, Apple Inc. released iPhone X with a facial recognition system called Face ID installed. Face ID allows user to unlock the phone by looking at the front camera of the phone (Savov, 2017).

On December 2016, Amazon launched a chain of convenience store named Amazon Go in the United States. The store concept uses facial recognition and various technologies to automate the purchase, checkout and payment steps associated with a retail transaction. Customers can purchase products without checking out at a cashier or self-checkout station (Garun, 2016).

On January 2019, the state of Penang launched a facial recognition system capable of detecting the faces of criminals. It uses Intelligent Video Analytics (IVA) to identify and track the location of the criminals. The camera will trigger alarm once the latest movement of the criminal is captured (Burt, 2019).

There are many more applications of facial recognition on improving business product, consumer experience and even advancing a nation's interest.

Challenge

The challenge of doing facial recognition on digital image is the difficulty of extracting facial features from pixels value. The recognition problem is made difficult by the variability of a face in the image. This includes variability in head rotation and tilt, light intensity and angle, facial expression, aging and other factors. (Bledsoe, Chan & Bisson, n.d.)

Traditional Facial Recognition Methods

The traditional face recognition algorithms can be categorized into holistic features and local appearance features like LBP. Holistic approach can be further categorized into linear projection approaches like linear discriminate analysis (LDA), principal component analysis (PCA), independent component analysis (ICA), linear regression classifier (LRC) and 2DPCA , and non-linear projection such as kernel PCA (KPCA), locally linear embedding (LLE), locality preserving projection (LPP) and self-organizing map (SOM).

In holistic approach, individual features like eyes, nose and mouth are ignored, and a complete face is considered as a single feature for detection and recognition. Both LDA and PCA focus on the global Euclidean Structure while LRC focuses on the local manifold structure. However, all these linear subspace learning algorithms-based methods fail to adequately represent faces when variations like illumination and facial expressions are present. Non-linear methods like KPCA and kernel ICA which use kernel techniques are introduced to overcome the drawback of traditional linear methods. However, these methods do not produce a significant improvement compared with linear methods. LLE and LPP which are capable to handle complex data and inherit the simplicity from linear approach have drawbacks too as they only projects for training data.

Local appearance approach has advantages over holistic approach. The approach is more stable to local changes like occlusion, misalignment and expression (Hassaballah & Aly, 2015). In facial recognition using local binary patterns (LBP), the face area is divided into smaller parts. Histograms are then extracted and integrated into a single feature vector. The representation of the face is formed using feature vector and similarities between images can be measured (Rahim, Hossain, Wahid & Azam, 2013). However, the recognition speed might be slowed down due to long histogram produced and the binary data produced are sensitive to noise (Fu & Wei, 2008).

Neural Network

Clearly, traditional face recognition methods are not the best approaches for image processing. Neural network which is capable of learning and modelling non-linear and complex relationships is introduced to overcome the drawback of traditional face recognition methods. Neural network is powered by massive amounts of data to learn a face representation that can cope with variations in the training data.

Convolutional Neural Network (CNN)

Convolutional neural network has been the state-of-the-art models for image recognition. Using a trained large and deep convolutional neural network, the model achieved a top-1 and top-5 error rates of 37.5% and 17.0% on the ImageNet LSVRC-2010 contest, which is significantly better than the previous state-of-the-art models (Krizhevsky, Sutskever & Hinton, 2012).

In convolutional neural network, a series of filters is applied to pixel data of an image to extract high-level features, which the model use for classification. CNN is comprised of three components:

- **Convolutional layers**, which apply convolutional filters to the image. For each subregion, convolution performs mathematical operations and produce a single value in the feature map as output. The layers then apply activation function on every value of feature map to introduce non-linearities.

- **Pooling layers**, which reduce the spatial size of the convolved feature in order to reduce the dimensionality of the feature map to decrease the processing time. There are two types of pooling, which are max pooling and average pooling. Max pooling removes the noisy activations together with the dimensionality reduction while average pooling performs dimensionality reduction as noise suppressant (Saha, 2018).
- **Fully connected layers**, which perform classification on the final feature maps (TensorFlow, n.d.).

Each CNN layer learns filter of increasing complexity. In the first layers, filters that detect basic features such as the edges of the image is learned. In the middle layers, filters that detect parts of objects is learned. For facial recognition, the layers learn to respond to face features like nose and eyes. In the last layers, the layers learn to recognize the full objects in various positions and shapes (Stewart, 2019).

MobileNetV2

There have been various CNN architectures and design that achieved state-of-the-art performance. However, the best CNN architectures in terms of accuracy are often difficult to train due to large number of parameters and number of operations. Comparisons were made to gauge the efficiency and effectiveness of various CNN architectures. MobileNetV2 achieves very good Top-1 accuracy versus floating-point operations (FLOPs) required tradeoff and has a relatively high Top-1 accuracy density (Bianco, Cadene, Celona & Napoletano, 2018).

In this project, we want to examine the performance of using MobileNetV2 CNN architecture on doing facial recognition.

MobileNetV2 is a new neural network architecture that is specifically tailored for mobile and resource constrained environments. It significantly reduces the memory required and number of operations while maintaining the same accuracy. It introduces a novel layer module which is an inverted residual with linear bottleneck. The module takes a low-dimensional compressed representation input which is first expanded to high dimension and filtered with a lightweight depth wise convolution. The features are then projected back to a low-dimensional representation with a linear convolution (Sandler, Howard, Zhu, Zhmoginov & Chen, 2019).

Objective

The objective of this project is to predict the identity, gender, and age group of a person from a given digital image of the person using convolutional neural network (CNN) model, specifically the MobileNetV2 architecture. This can be achieved by training three separate CNN models using large number of images data to predict the age group, gender, and identity of a person.

The first CNN model will be trained on the Labelled Faces in the Wild (LFW) deep funneled images (Huang, Mattar, Lee & Miller, 2012). The dataset contains 13,233 images of 5,749 people detected and centered by the Viola Jones face detector and collected from the web. Each image is labelled. This allows a model to extract important facial features to correctly identify the person in the image.

The second CNN model will be trained on the UTKFace dataset to predict the gender of a person inside an image. This dataset is a large-scale face dataset containing 23,708 images, with each image labelled by age, gender and ethnicity.

The third CNN model will be trained on the same UTKFace dataset to predict the age group of a person inside an image. The labelled age for each image will be categorized into 10 age groups, namely age of 1 to 3, 4 to 6, 7 to 12, 13 to 18, 19 to 25, 26 to 35, 36 to 45, 46 to 60, 61 to 75, and 75 and above.

Finally, all these 3 CNN models will be used to classify data unseen by the model during training. There will also be an attempt to use the pre-trained identity model to do person re-identification by classifying the identity of a person using images of classes or identities not known to the model via feature extractions on CNN layer and clustering technique, without re-training the model.

Scenario and Methodology

Software and Hardware Dependencies

This project was run on the following software and package dependencies:

1) Windows 10 Pro (Version 1803, OS build 17134.765)
2) RStudio (Version 1.1.463)
3) R (Version 3.5.3, 64-bit)
4) R packages:
    a) caret 6.0-84
    b) dplyr 0.7.6
    c) ggplot2 3.1.1
    d) keras 2.2.4.1.9001
    e) KODAMA 1.5
    f) reticulate 1.12.0-9000
    g) Rtsne 0.15
    h) shiny 1.3.2
    i) tensorflow 1.13.1.9000
5) Anaconda3-2018.12-Windows-x86_64
6) Python 3.7.1
7) Python package:
    a. opencv-python==4.1.0.25
8) CUDA (Version 10.1.105_418.96)

This project was run on the following hardware:

1) Processor: Intel® Core™ i5-7400 CPU @ 3.00GHz
2) GPU: ASUS ROG Strix GeForce® GTX 1070 OC edition 8GB GDDR5
3) RAM: 8.00 GB DDR4

1. Identity Prediction

Data Acquisition

The ‘faces_data_new’ (FDN) was downloaded at *https://www.kaggle.com/gasgallo/faces-data-new* and the ‘Labeled Faces in the Wild’ (LFW) was downloaded at *http://vis-www.cs.umass.edu/lfw/#deepfunnel-anchor*.

For the LFW dataset, the deep funneled dataset is used because the data quality is better data quality.

Data Pre-processing

The LFW dataset contains 13,233 images of 5,749 people or classes. However, most of the class contains only 1 image. Having more images per class allow the model to generalize better on extracting the features of a specific person.

The LFW Dataset was filtered such that there are at least 15 images per class. Images were copied to separate folders, where each folder represents each class. All filtered images were cropped to include only the face using OpenCV Haar Cascade face detection algorithm. This is done using the “reticulate” package in R, which allows the use of Python OpenCV functions. Then, for each class, 10% of the images were extracted as an unseen test dataset.

Similarly, the FDN dataset was cropped and split into train and test dataset. All classes were included since most classes have at least 10 images.

Both datasets were combined as a bigger dataset with 9,935 images and 487 classes.

Data Loading and Image Augmentation

For scalability purposes, data were not loaded into RAM at the beginning. Instead, using Keras built-in functionalities, images can be loaded and trained by batches by specifying the directory of the training dataset.

Images loaded were passed through an image augmentation function which randomly rotates the image by up to 45 degrees, shifts the image vertically or horizontally, or does a horizontal flip of the image.

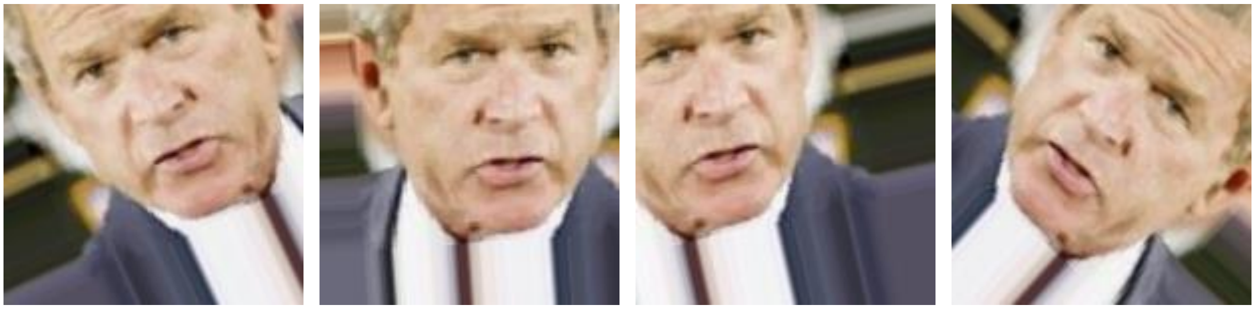


Figure 1: Examples of image augmentation

Image augmentation is important as an augmented image has the same label. Whether the face was rotated, shifted or flipped, the augmented face still belongs to the same person. Therefore, image augmentation allows the model to generalize and extract important features of the face instead of being overfitted to features that are irrelevant in predicting the identity of a person, such as light intensity of the image, specific pose of the face (e.g. Person A having a lot of images with the head tilted to the left, or exhibit a particular facial expression) or background of the image etc. Besides, by doing image augmentation, the same image could go through infinitely many augmentation configurations. This significantly increase the amount of data we have and improve the accuracy of the model. The output of an augmented image is a three-dimensional 150 by 150 by 3 (150, 150, 3) vector. The first dimension is the width, the second dimension is the height and the third dimension is the channels of the image.

Model Training

The dataset is split into training and validation dataset, where the proportion is 80% and 20% respectively. The augmented images in the form of (150, 150, 3) vectors were passed to the CNN model in batches of 32. The weights or parameters of the CNN models were randomly initialized by default. The input passed through the CNN layers.

The final layer of the MobileNetV2 CNN model is removed and replaced with 4 layers, namely a 2D global average pooling layer, a fully connected 128 neurons layer with rectified linear unit (ReLU) activation function, followed by a fully connected 256 neurons layer with ReLU activation function and an output layer of 487 neurons with softmax activation function.

The purpose of the 2D global average pooling layer is to condense the output of the 156th layer of the MobileNetV2 CNN model to a 1D vector. The added two dense layers act as the final feature extraction layers before predicting the classes of an image using the output layer with softmax activation function.

Since, this is a classification problem with 487 classes, the loss in this case is set to categorical cross-entropy. The formula for categorical cross-entropy is

$$-\sum_{c=1}^{M} y_{o,c} \log(p_{o,c})$$

, where M is the number of classes, log is the natural log, y is the binary indicator (0 or 1) if the class label c is the correct classification for observation o and p is the predicted probability observation o is of class c.

The model is trained iteratively for 100 epochs by adjusting the weights of the model through minimizing the loss using the RMSProp optimizer with learning rate of 0.0001.

Model Selection

For each epoch of training, the model weights were captured as checkpoint. The best model is selected by picking the checkpoint with the lowest validation loss. At this checkpoint, the model has the best fit, where it does not underfit nor overfit on the training dataset and performs poorly on the validation dataset.

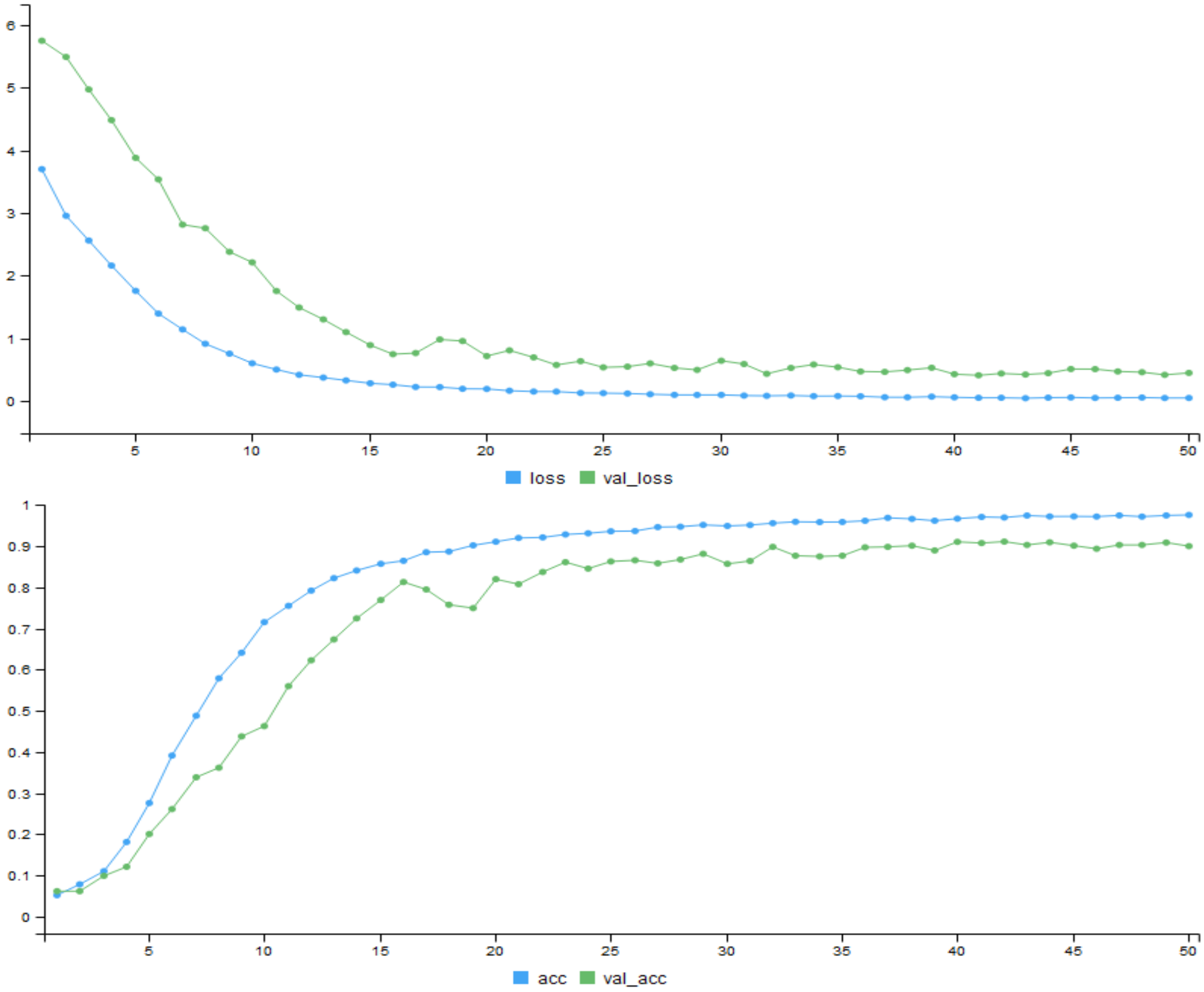


Figure 2: Learning curves for identity prediction

In this case, the best model is at the 41st epoch. The accuracy of this trained model is then evaluated by comparing the predictions made on the unseen test dataset and the respective true label.

2. Gender Prediction

Data Acquisition

The same UTKFace dataset used for gender prediction was used to do age group prediction. The dataset was downloaded at https://susanqq.github.io/UTKFace/. The aligned and cropped faces dataset was used.

Data Pre-processing

The UTKFace dataset contains 23,708 images. The images were cropped to include the face only by default. The only pre-processing step was to copy the images to two different folders labelled 0 for male, and 1 for female.

Data Loading and Image Augmentation

Images loaded were passed through an image augmentation function which randomly rotates the image by up to 30 degrees, shifts the image vertically or horizontally, or does a horizontal flip of the image.

The output of an augmented image is a three-dimensional 128 by 128 by 3 (128, 128, 3) vector. The first dimension is the width, the second dimension is the height and the third dimension is the channels of the image.

Model Training

The dataset is split into training and validation dataset, where the proportion is 80% and 20% respectively. The augmented images in the form of (128, 128, 3) vectors were passed to the CNN model in batches of 32. The weights or parameters of the CNN models were randomly initialized by default. The input passed through the CNN layers.

The final layer of the MobileNetV2 CNN model is removed and replaced with 2 layers, namely a 2D global average pooling layer, and an output layer of 2 neurons with softmax activation function.

Since, this is a classification problem with just 2 classes, male or female, the loss in this case is set to categorical cross-entropy, which is equivalent to the binary cross-entropy.

The model is trained iteratively for 30 epochs by adjusting the weights of the model through minimizing the loss using the RMSProp optimizer with learning rate of 0.00002.

Model Selection

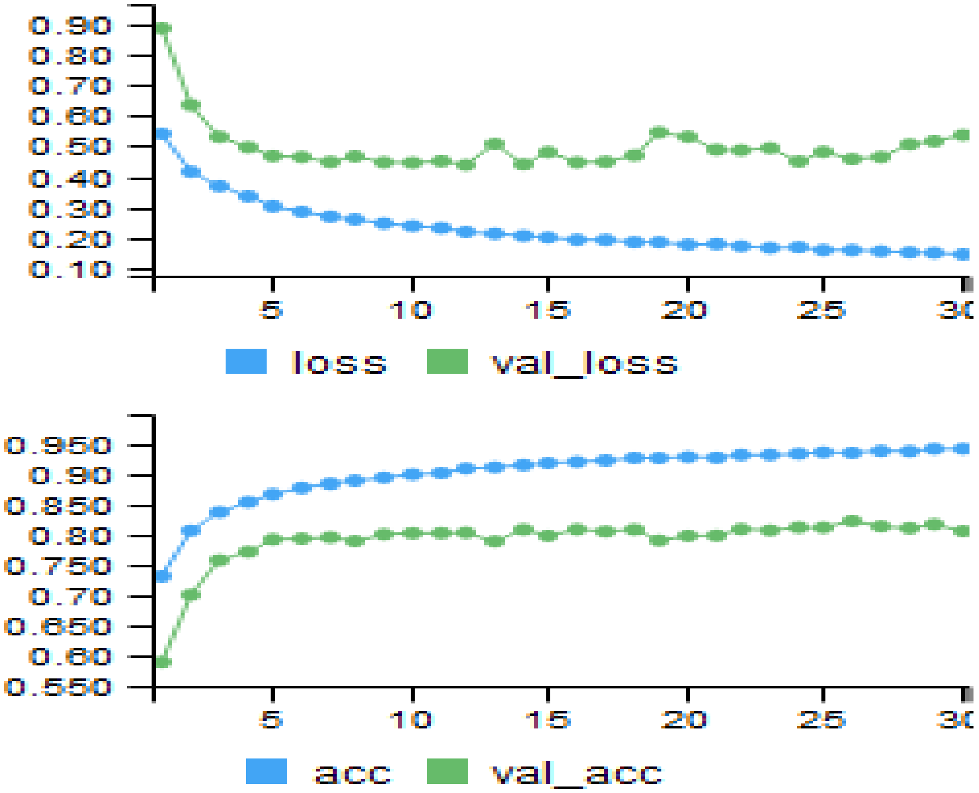


Figure 3: Learning curves for gender prediction

The best model is selected on the lowest validation loss. In this case, the model achieved the best performance at the 12th epoch. The model is then evaluated on the test dataset.

3. Age Group Prediction

Data Acquisition

The same UTKFace dataset used for gender prediction was used to do age group prediction. The dataset was downloaded at https://susanqq.github.io/UTKFace/. The aligned and cropped faces dataset was used.

Data Pre-processing

The UTKFace dataset contains 23,708 images. The images were cropped to include the face only by default. For the entire dataset, the age spans from 1 to 116 years old. Due to functionality constraint in Keras for R, we could only do classification on the age target. Therefore, the age was divided into 10 age groups, namely age of 1 to 3, 4 to 6, 7 to 12, 13 to 18, 19 to 25, 26 to 35, 36 to 45, 46 to 60, 61 to 75, and 75 and above. 10 folders were created to represent each age group. Each image in the dataset was copied to the respective directory.

Data Loading and Image Augmentation

Images loaded were passed through an image augmentation function which randomly rotates the image by up to 30 degrees, shifts the image vertically or horizontally, does a horizontal flip of the image, changes the brightness of the image, performs channel shift, or zooms the image by a random factor.

The output of an augmented image is a three-dimensional 128 by 128 by 3 (128, 128, 3) vector. The first dimension is the width, the second dimension is the height and the third dimension is the channels of the image.

Model Training

The dataset is split into training and validation dataset, where the proportion is 80% and 20% respectively. The augmented images in the form of (128, 128, 3) vectors were passed to the CNN model in batches of 32. The weights or parameters of the CNN models were randomly initialized by default. The input passed through the CNN layers.

The final layer of the MobileNetV2 CNN model is removed and replaced with 2 layers, namely a 2D global average pooling layer, and an output layer of 10 neurons with softmax activation function.

Since, this is a classification problem with just 10 classes, namely the 10 age groups, the loss in this case is set to categorical cross-entropy.

The model is trained iteratively for 50 epochs by adjusting the weights of the model through minimizing the loss using the RMSProp optimizer with learning rate of 0.00003.

Model Selection

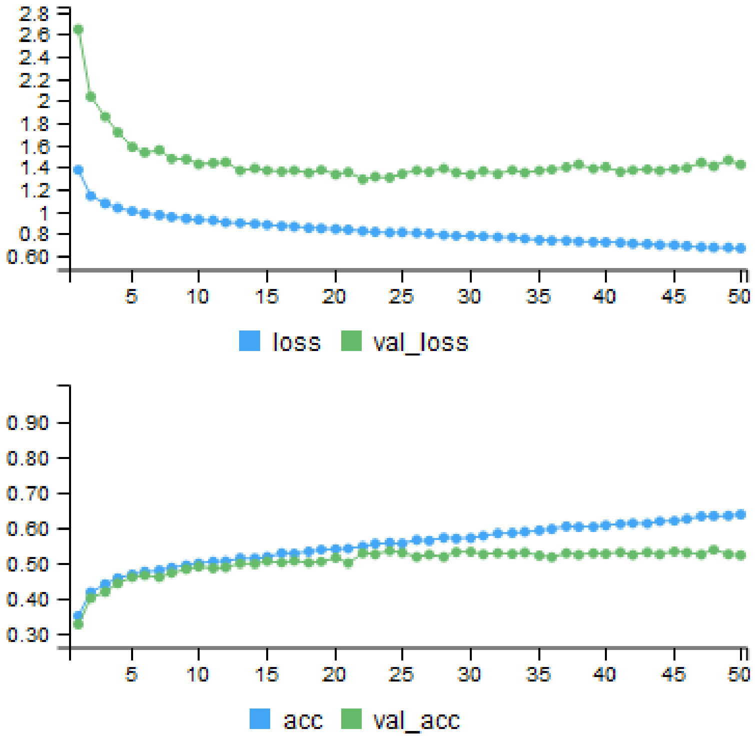


Figure 4: Learning curves for age group prediction

The best model is selected on the lowest validation loss. In this case, the model achieved the best performance at the 22nd epoch. The model is then evaluated on the test dataset.

4. Person Re-Identification using Pre-trained Model

Data Acquisition

We used the same LFW Dataset as mentioned in identity prediction for this section.

Data Pre-processing

Previously, the LFW Dataset was filtered such that there are at least 15 images per class. For this section, we filtered classes that have at least 12 images but less than 15 images. These images were totally unseen by the model trained for identity prediction. This dataset was divided into 80% train dataset and 20% test dataset.

The unseen train dataset has 327 images and 31 classes; the unseen test dataset has 61 images and 31 classes.

The train dataset used to train the identity model was used as well. This train dataset has 9,935 images and 487 classes.

Data Loading

Three different datasets will be used to do the person re-identification. The following is a self-defined term for each dataset:

1) Train dataset is the dataset used to train the identity model. It contains 9,935 images and 487 classes.
2) Unseen train dataset is the 80% training dataset extracted from the unseen dataset. It is used to generate feature clusters for person re-identification on the unseen test images. This dataset contains 327 images and 31 classes, which the classes do not exist under the train dataset.

3) Unseen test dataset is the 20% testing dataset extracted from the unseen dataset. It is used to evaluate the accuracy of this person re-identification technique. It contains 61 images and 31 classes.

No data augmentation is performed, these datasets will be passed to the pre-trained identity model to extract the features in each image.

Features Extraction using Pre-trained Identity Model

The previously trained identity model is loaded. In this section, we want to extract the important features of a face image. Therefore, the features output we needed is the output at the 157th layer of the CNN models we used earlier.

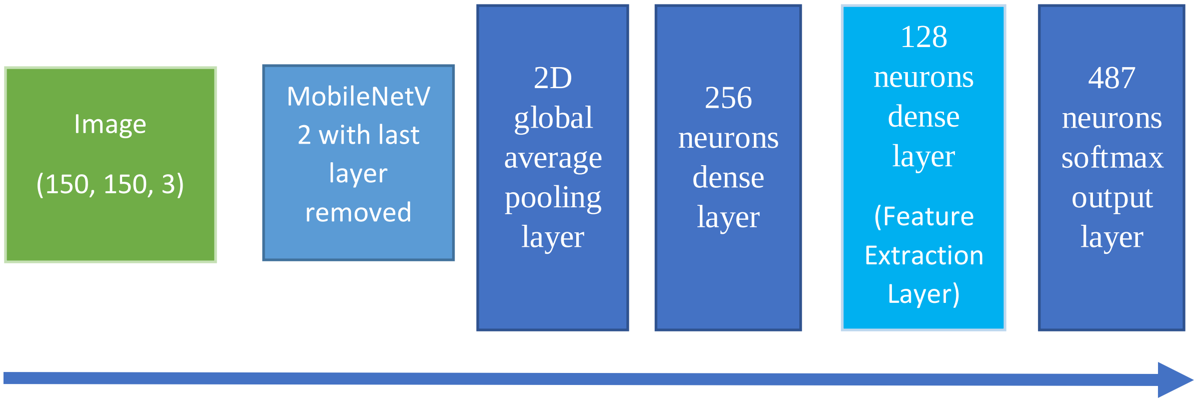


Image
(150, 150, 3)
MobileNetV 2 with last layer removed
2D global average pooling layer
256 neurons dense layer
128 neurons dense layer
(Feature Extraction Layer)
487 neurons softmax output layer

Figure 5: Flow of image through the identity CNN model and the 128 neurons feature extraction layer

The features extracted is a 128-length vector per image.

We performed the features extraction on both the train dataset and unseen train dataset. Then, using K-nearest neighbor, we predicted the classes on the unseen test dataset based on the closeness of unseen test dataset extracted features and the features extracted from the train and unseen train dataset.

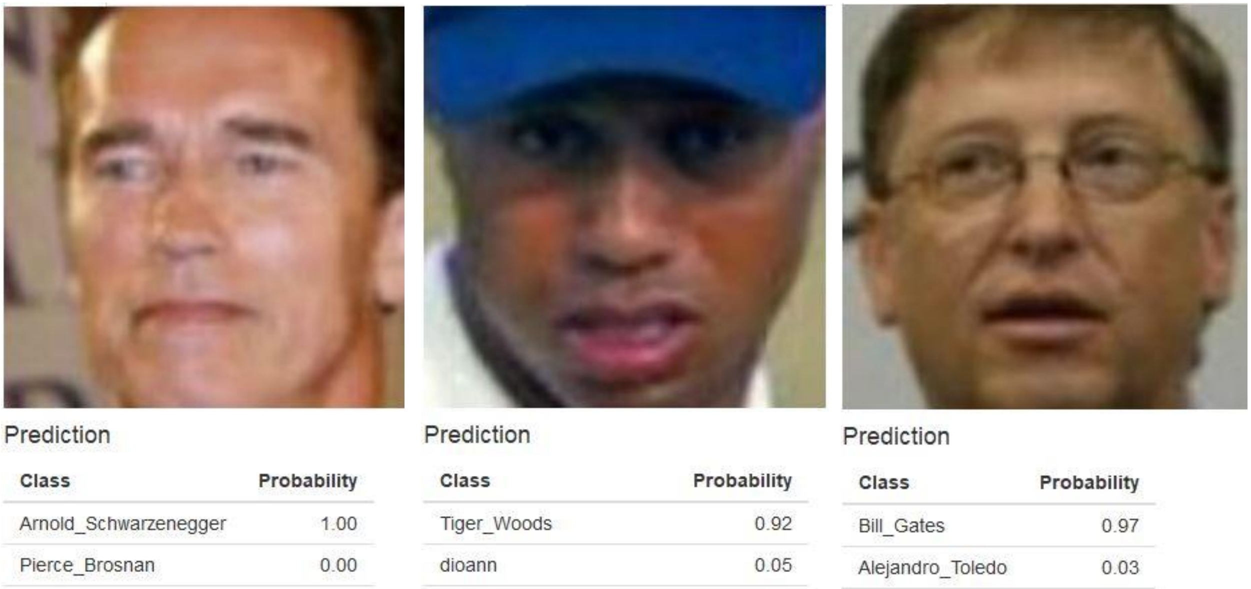
K was set to 7. Based on the majority system, the final predicted class is taken as the mode of the nearest 7 classes based on the features extracted.

The features extracted were then visualized on a 2-dimensional scatterplot by plotting the 2D features extracted using t-distributed stochastic neighbor embedding (t-SNE) algorithm.

Results

Results: Identity Prediction

The model achieved a training, validation and testing accuracy of 99.29%, 93.71% and 94.14% respectively. This is an acceptable accuracy considering that we trained the model on a consumer grade GPU in less than 2 hours.



Prediction

Class	Probability
Arnold_Schwarzenegger	1.00
Pierce_Brosnan	0.00

Prediction

Class	Probability
Tiger_Woods	0.92
dioann	0.05

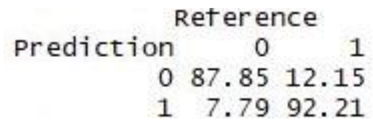
Prediction

Class	Probability
Bill_Gates	0.97
Alejandro_Toledo	0.03

Figure 6: Identity predictions samples

Results: Gender Prediction

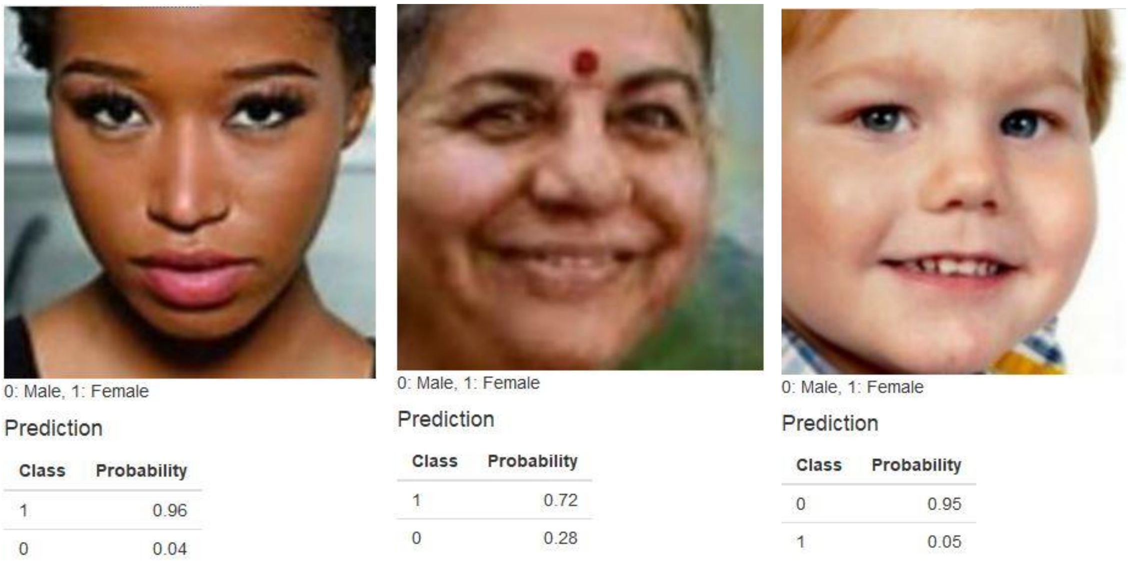
The model achieved an overall training, validation and testing accuracy of 93.04%, 82.47% and 89.79% on gender prediction. Gender seems to be a harder target to predict compared to identity because of the great variability of facial features for a gender. Besides, there may be overlapping features for both gender that further increase the difficulty to distinguish correctly the gender of a person.



```
              Reference
Prediction      0     1
         0 87.85 12.15
         1  7.79 92.21
```

Figure 7: Confusion matrix for gender prediction (in percentage)

The figure above shows the confusion matrix in percentage terms on the test dataset that describes the accuracy of the model on predicting the gender. The model seems to perform slightly worse on predicting male (class 0) than female (class 1).



0: Male, 1: Female

Prediction

Class	Probability
1	0.96
0	0.04

0: Male, 1: Female

Prediction

Class	Probability
1	0.72
0	0.28

0: Male, 1: Female

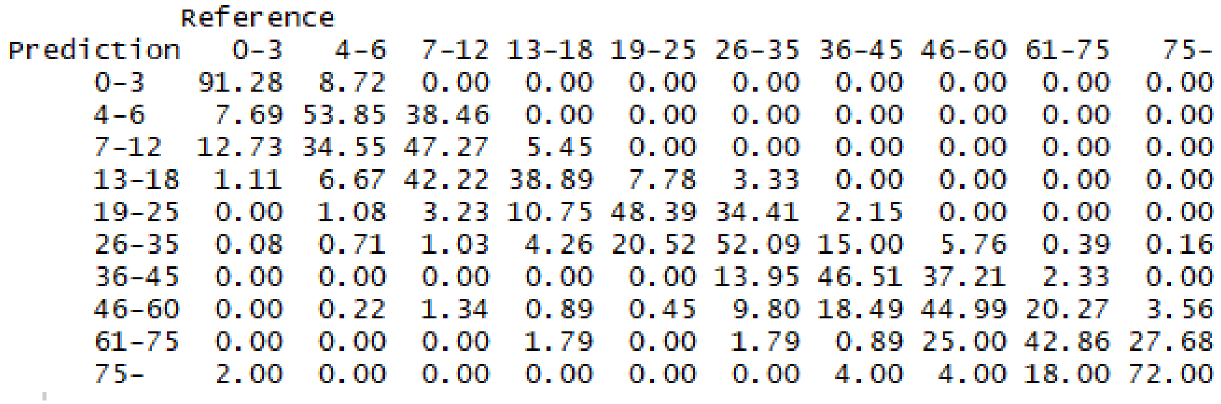
Prediction

Class	Probability
0	0.95
1	0.05

Figure 8: Gender predictions samples

Results: Age Group Prediction

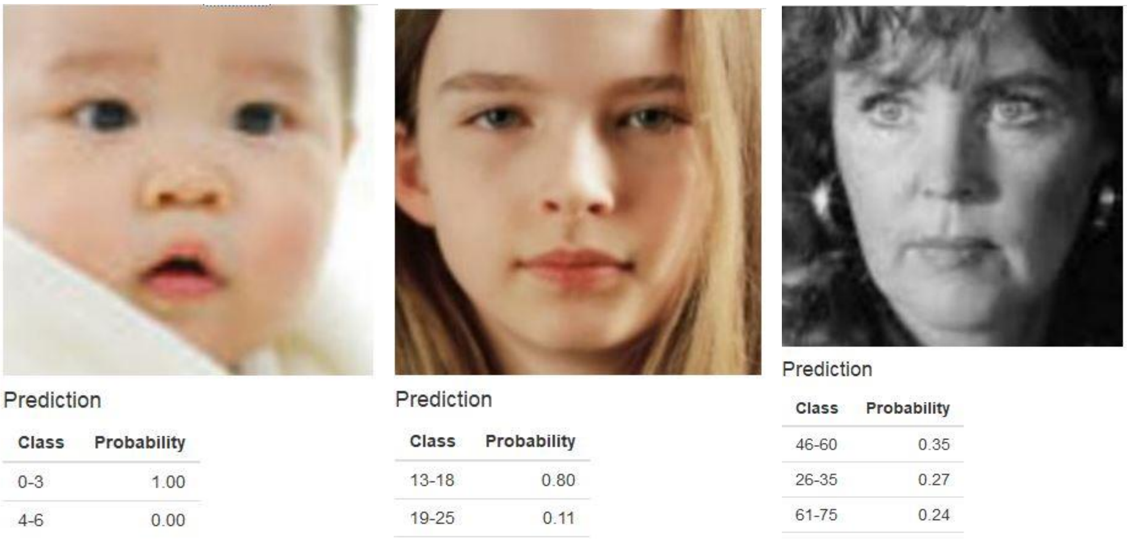
The model achieved an overall training, validation and testing accuracy of 93.04%, 82.47% and 89.79% on age group prediction.



```
          Reference
Prediction   0-3   4-6  7-12 13-18 19-25 26-35 36-45 46-60 61-75   75-
     0-3   91.28  8.72  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00
     4-6    7.69 53.85 38.46  0.00  0.00  0.00  0.00  0.00  0.00  0.00
     7-12  12.73 34.55 47.27  5.45  0.00  0.00  0.00  0.00  0.00  0.00
     13-18  1.11  6.67 42.22 38.89  7.78  3.33  0.00  0.00  0.00  0.00
     19-25  0.00  1.08  3.23 10.75 48.39 34.41  2.15  0.00  0.00  0.00
     26-35  0.08  0.71  1.03  4.26 20.52 52.09 15.00  5.76  0.39  0.16
     36-45  0.00  0.00  0.00  0.00  0.00 13.95 46.51 37.21  2.33  0.00
     46-60  0.00  0.22  1.34  0.89  0.45  9.80 18.49 44.99 20.27  3.56
     61-75  0.00  0.00  0.00  1.79  0.00  1.79  0.89 25.00 42.86 27.68
     75-    2.00  0.00  0.00  0.00  0.00  0.00  4.00  4.00 18.00 72.00
```

Figure 9: Confusion matrix for age group prediction (in percentage)

The figure above shows the confusion matrix in percentage terms on the test dataset that describes the accuracy of the model on predicting each age group. We can observe that the model has a difficulty predicting the exact age group but is able to predict into a nearer age group. It has some "broad sense" of age and rarely predict very inaccurate age group. For example, the model does not predict a person age 36 and above into the age group of "13-18".



Prediction

Class	Probability
0-3	1.00
4-6	0.00

Prediction

Class	Probability
13-18	0.80
19-25	0.11

Prediction

Class	Probability
46-60	0.35
26-35	0.27
61-75	0.24

Figure 10: Age group predictions samples

Results: Person Re-Identification

The accuracy on unseen test data is 50.82%. It shows that the model is unable to extract enough important facial features from limited amount of unseen test images to correctly identify the unseen test classes. However, the accuracy shows that the model is not completely random, and there are still room for improvements using state-of-the-art technique such as Discriminatively Learned CNN Embedding (Zheng, Zheng & Yang, 2017).

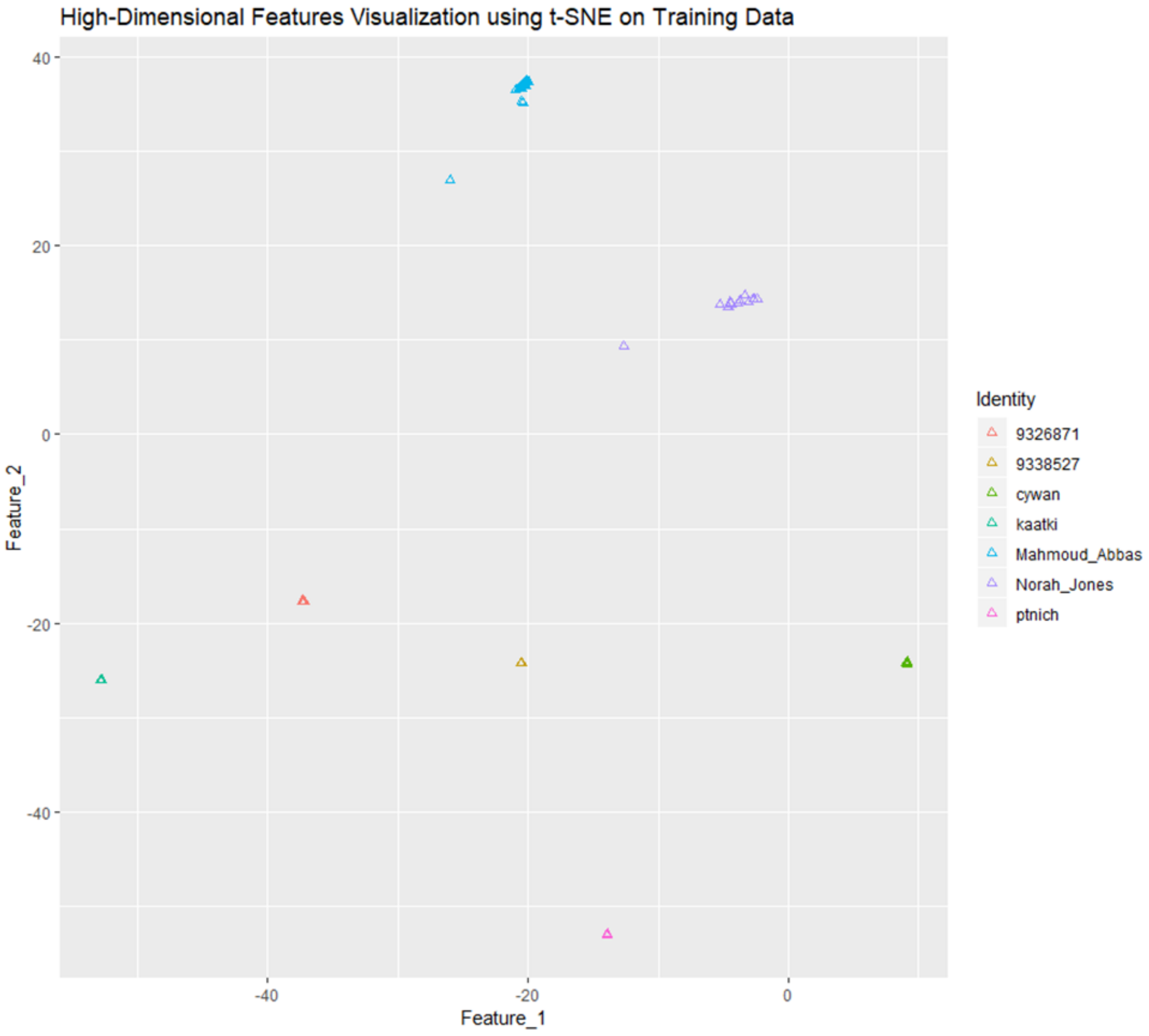


Figure 11: Features Visualization on Train Data (sample of 7 classes)

From the figure above, we can observe that the feature clusters formed are highly distinguishable by their respective class labels. It shows that our model can extract unique features on the training images to classify the respective classes or identities.

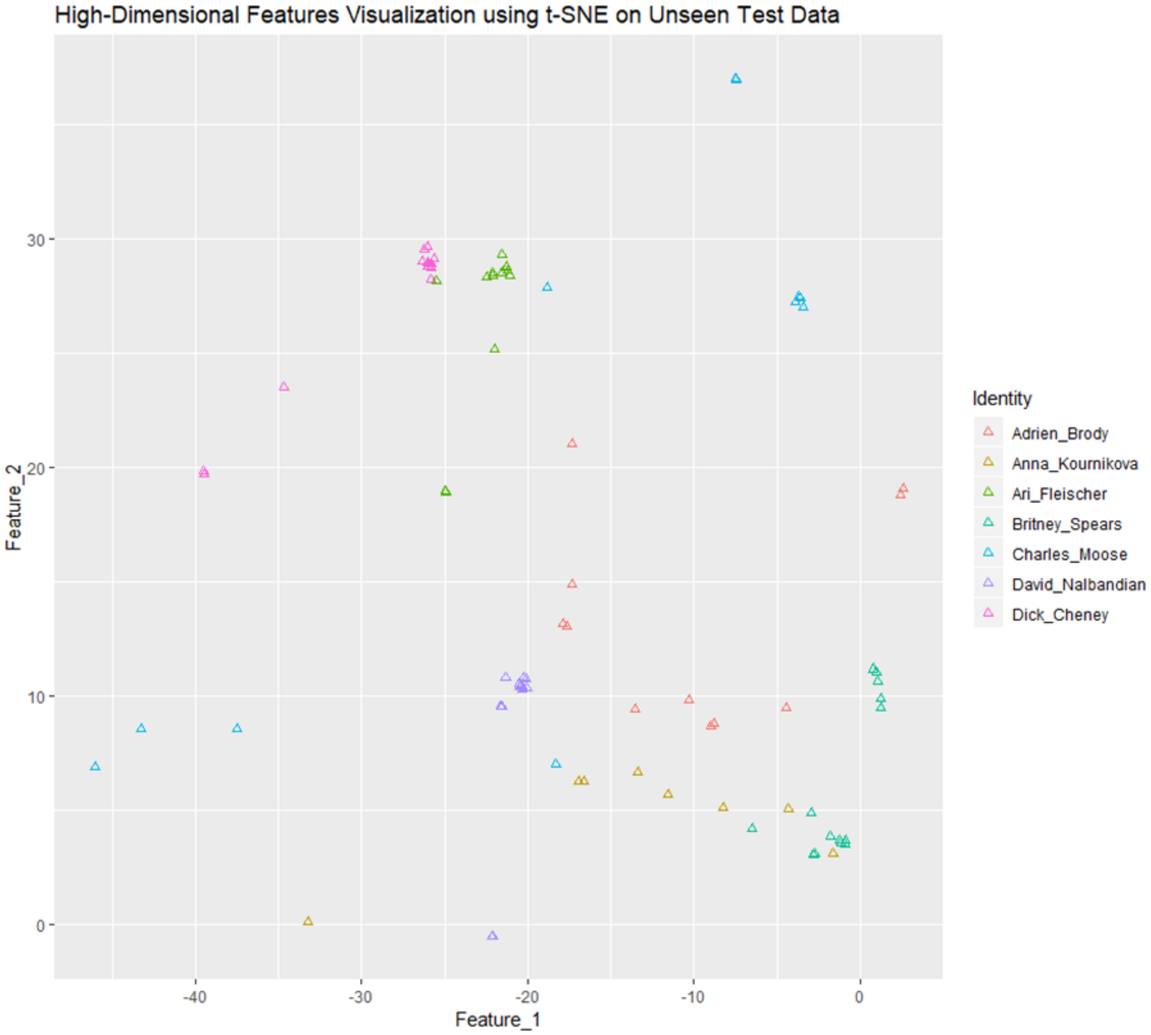


Figure 12: Features Visualization on Unseen Test Data (sample of 7 classes)

From the figure above, we can observe that some of the feature clusters formed can be distinguished by their respective class labels. It shows that our model could not extracts unique facial features to classify the respective classes or identities of unseen test images.

Results: Accuracies Summary

No	Prediction Type	Training	Validation	Testing
1	Identity	99.29%	93.71%	94.14%
2	Gender	93.04%	82.47%	89.79%
3	Age Group	61.00%	53.55%	53.10%
4	Person Re-Identification	-	-	50.82%

Table 1: Accuracies for each prediction task

Conclusion

This project achieved the aim to perform facial recognition using convolutional neural network (CNN), specifically MobileNetV2 architecture, trained on various large dataset of face images. Three separate models were trained to perform predictions on identity, gender and age group of a person given the image of the face of the person.

The steps in training these three models involved data acquisition, data pre-processing, data loading and image augmentation, model training and model selection.

An attempt to do person re-identification using pre-trained identity model is also performed. Facial features were extracted at the intermediate layers of the CNN models for each image. These facial features were treated as basis to form features cluster at a high-dimensional space. K-nearest neighbor algorithm is used to classify unseen test dataset by measure of closeness between the extracted features of the unseen test dataset and the features cluster of the train dataset and unseen train dataset.

Using a consumer grade GPU, the models achieved satisfactory accuracies on predicting the identity, gender and age group as detailed in Table 1. There are room for improvements on performing person re-identification.

References

Bledsoe, W. W., Chan, H. & Bisson, C. (n.d.). *Woodrow Bledsoe Originates of Automated Facial Recognition*. Retrieved from http://www.historyofinformation.com/ detail.php?entryid=2495

Bianco, S., Cadene, R., Celona, L. & Napoletano, P. (2018). Benchmark Analysis of Representative Deep Neural Network Architectures. *IEEE Access*, 4(0). Retrieved from https://arxiv.org/pdf/1810.00736.pdf

Burt, C. (2019). Malaysian state launched facial recognition to CCTV network. Retrieved from https://www.biometricupdate.com/201901/malaysian-state-launches-facial-recognition-to-cctv-network

Fu, X. & Wei, W. (2008). Centralized Binary Patterns Embedded with Image Euclidean Distance for Facial Expression Recognition. *ICNC '08 Proceedings of the 2008 Fourth International Conference on Natural Computation*, 4(0), 115-119. Retrieved from https://dl.acm.org/citation.cfm?id=1473751

Garun, N. (2016). Amazon just launched a cashier-free convenience store. Retrieved from https://www.theverge.com/2016/12/5/13842592/amazon-go-new-cashier-less-convenience-store

Hassaballah, M. & Aly, S. (2015). Face Recognition: Challenges, Achievements, and Future Directions. IET Comput. Vis., 9(4), 614-626. Retrieved from https://www.researchgate.net/publication/271584966_Face_Recognition_Challenges_Achievements_and_Future_Directions

Huang, G. B., Mattar, M., Lee, H. & Miller, E. L. (2012). Learning to Align from Scratch. *Neural Information Processing Systems (NIPS)*. Retrieved from https://papers.nips.cc/paper/4769-learning-to-align-from-scratch

Krizhevsky, A., Sutskever, I. & Hinton G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. *Proceedings of the 30th International Conference Machine Learning*. doi: 10.1145/3065386. Retrieved from https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

Md. Abdur Rahim, Md. Najmul Hossain, Wahid, T. & Md. Shafiul Azam. (2013). Face Recognition using Local Binary Patterns (LBP). *Global Journal Of Computer Science And Technology Graphics & Vision*, 13(4). Retrieved from https://globaljournals.org/GJCST_Volume13/1-Face-Recognition-using-Local.pdf

Saha, S. (2018). A Comprehensive Guide to Convolutional Neural Networks – the ELI5 way. Retrieved from https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A. & Chen, L. (2019). *MobileNetV2: Inverted Residuals and Linear Bottlenecks*. Retrieved from https://arxiv.org/pdf/1801.04381.pdf

Savov, V. (2017). iPhone X announced with edge-to-edge screen, Face ID, and no home button. Retrieved from https://www.theverge.com/2017/9/12/16288806/apple-iphone-x-price-release-date-features-announced

Stewart, M. (2019). Simple Introduction to Convolutional Neural Networks. Retrieved from https://towardsdatascience.com/simple-introduction-to-convolutional-neural-networks-cdf8d3077bac

TensorFlow. (n.d.). Build a Convolutional Neural Network using Estimators. Retrieved from https://www.tensorflow.org/tutorials/estimators/cnn

Zheng, Z., Zheng, L. & Yang, Y. (2017). A Discriminatively Learned CNN Embedding for Person Re-identification. Retrieved from https://arxiv.org/abs/1611.05666