**ML Project – Face Recognition Attendance System**

**Group Members**:

**Arthur Vincent Chin (101218817)**

**George Kennedy (101218969)**

**Raphael Jong Jun Jie (101228531)**

**Masrur Rahman Zahin (101214608)**

# 1 Methodology

The aim of the project is to create an attendance system that uses a face recognition feature to capture the faces on an employee and take their attendance. To do this, a face detection model needs to be trained first so that the program can use it to detect and verify that the person checking in is in fact an employee otherwise the program should identify them as an unknown person.

For creating and training the face recognition models, two methods will be used which are classification method which aims identifying the category of a new observation amongst a set of categories on the basis of a labelled training set, and the deep metric learning method which uses neural networks to automatically learn discriminative features from images and then compute the metric.

## 1.1 MobileNet

For the classification method, the model chosen to perform face recognition training on is the MobileNet model. MobileNet is built on depthwise separable convolutions except for the first layer which is a full convolutional layer. All of it layers are followed by batch normalization and ReLU non-linearity and the final layer is a fully connected layer without any non-linearity and feeds to the softmax for classification. For the down sampling, strided convolution is used for both depthwise convolution as well as for the first fully convolutional layer. In total, there is 28 layers for MobileNet considering depthwise and pointwise convolution as separate layers (Nepal 2020). The model architecture can be seen in Figure *2*.

## 1.2 Multi-Class Classification

For multi-class classification method, the model was built from ground up with the full dataset provided. The model contained four convolutional layers, and 3 dense layers. Each convolutional layer was followed by Batch Normalization and Max Pooling and used ReLU as an activation function. The first layer is a full convolutional layer, and the last layer is a fully connected dense layer and Softmax activation function was used for the multi-class classification.

## 1.3 Siamese Network with Contrastive Loss

For the metric learning method, the Siamese Network was built with 2 identical CNN networks called Embedding Networks sharing the same weights and are responsible for producing vector representations for the inputs. The vector representations are to be compared by Euclidian Distance layer in order to calculate the distance similarity between vectors which is then optimized by Contrastive Loss (Bekuzarov 2020). The margin for the loss was set to 1 which indicates the similarity between two images. The closer the loss value to 1 the more similar both images are. Also, the threshold of 0.5 was set to the prediction, if value of prediction is more than or equal to 0.5, the model indicates both images are of the same person. The models architecture can be seen in **Siamese Network Architectures**.

# 2 Result and Discussion

## 2.1 MobileNet

For the MobileNet model, it was trained on a reduced data that consists of 16, 095 and 400 images for train and test respectively which belongs to 200 classes. The first training sequence was done with some of the base model’s layers unfrozen because the dataset used is a small dataset and is different from the pre-trained model’s dataset and the loss function used was categorical cross entropy and the optimizer used was RMSprop with a learning rate of 0.001. The batch size was set to 16 and the additional layers created were a global average pooling2D layer, two dense layers with 512 neurons for both layers, and a prediction layer with 200 neurons to represent the 200 classes in the reduced dataset. After training was done, the model performed below average with an accuracy of 74%, a loss of 0.91, and an ROC AUC score of 0.68.

After the first training sequence, the model was then finetuned to freeze all the previous layers until a specific cut off point and leave the rest of the layers trainable which will help in improving the model’s accuracy due to the previously trained weights learning the features of the dataset. The model was then trained with the neurons in the two dense layers hyper tuned to 256 neurons for both layers, and with batch size of 32. The final finetuned model was also trained for another 13 epochs in total before the validation loss plateaued and early stopping was used to stop the training at that point. The model could still do better therefore it was trained longer for another 5 epochs and the final performance of the MobileNet model gave an accuracy of 80%, a loss of 0.70, and an ROC AUC score of 0.75.

## 2.2 Multi-Class Classification

The model was trained with a data consisting of 8000 images belonging to 4000 classes for all Train, Validation and Testing. The training was done 3 separate times each time with 50, 70, 100 epochs with a batch size of 32 and weights was saved in an h5 file. The model that provided the best accuracy was chosen which was 70 epochs. After the training was completed, the model was later evaluated, and it provide an accuracy of 71%, loss of 1.07.

## 2.3 Siamese Network with Contrastive Loss

The Siamese Network was tested on 3 different CNN architectures (Embedding Networks) built from scratch and only the best model chosen. The dataset used for training was the same as the MobileNet model’s above. The images were then duplicated to make pairs of images for the inputs so that in total 32,190 & 800 images used for training & validation respectively. The input shape of 100 by 100 pixels with 3 channels were set for each embedding nets.

For Siamese Network with Embedding Net 1, the model architecture built was simple. it consisted of the input shape passed into Batch Normalization at the first layer, 2 Conv2D layers with 4 & 16 filters respectively, AveragePooling2D, Flatten and Dense layer with 200 neurons. The TanH activation function was used for the Conv2D & Dense layers. Then, this Embedding Net passed into the custom Euclidean Distance layer to be then received by the output layer which is the Dense layer with 1 neuron and Sigmoid activation, whereby it was used for the binary classification of determining whether both images belong to the same person (1) or not (0). Finally, by passing the output layer as the output of a new keras model and the input shape as its input, the Siamese Network had successfully created. Next, the network was compiled using RMSprop opitmizer with 0.001 learning rate and the custom Contrastive Loss. and then trained with the batch size of 32. After several epochs, the model’s accuracy growth was stagnant and finally gave an accuracy of 69%, a loss of 0.20 and an ROC AUC score of 0.85.

For Siamese Network with Embedding Net 2, the model architecture was similar to the Embedding Net 1, the differences were this network used ReLU activation for the Conv2D & Dense layers, and compiled the Siamese model using Adam optimizer. The model was stopped from training as it became overfitting. Finally, the model gave an accuracy of 87%, a loss of 0.09 and an ROC AUC score of 0.87.

For Siamese Network with Embedding Net 3, the embedding architecture was built with more complex networks according to Koch, Zemel & Salakhutdinov (n.d.) paper with some adjustments. It consisted of 3 Conv2D layers with 128, 128, 256 filters respectively, Batch Normalization & MaxPooling layers. There were also Dropout layers added in to minimize the overfitting, and for the final layer, a Dense with 4096 neurons Sigmoid activation was implemented. Next, this embedding net went through the same process as mention above to form the Siamese Network. It was compiled with Adam optimizer with 0.001 learning rate and the custom Contrastive Loss. After several epochs, the training was stopped to prevent overfitting, which resulted in 80% accuracy, 0.14 loss, and 0.91 ROC AUC Score. This Siamese Network is the best among other two Siamese Nets as the accuracy and loss are balanced.

## 2.5 Model Comparison

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **Accuracy** | **Loss** | **ROC AUC Score** |
| MobileNet Finetuned | 74% | 0.91 | 0.73 |
| MobileNet Finetuned Final | 80% | 0.70 | 0.75 |
| Siamese Network with Contrastive Loss (Embedding Nets 1) | 69% | 0.20 | 0.85 |
| Siamese Network with Contrastive Loss (Embedding Nets 2) | 87% | 0.09 | 0.87 |
| Siamese Network with Contrastive Loss (Embedding Nets 3) | 80% | 0.14 | 0.91 |
| Multiclass Classification Model | 71% | 1.07 | - |

## 2.6 Face Attendance System

The model chosen to be used in the face attendance program is the MobileNet model. For the face attendance system, the user interface will be deployed on Streamlit which is a browser application that will run the program’s graphical user interface (GUI). The screenshots of the program can be seen in *Figure 13* to *Figure 16*.

# 3 Presentation Video and GitHub Repo Link

This is the YouTube link to the video presentation and GitHub repository link:

Presentation Video: https://www.youtube.com/watch?v=\_jhwsFip9xw

GitHub Repository: https://github.com/COS30082-ML-Project/ML-Project

# 4 Appendix

Diagram, table

Description automatically generated

Figure 1: Standard convolution with batchnorm and relu

A picture containing text, electronics

Description automatically generated

Figure 2: MobileNet architecture

## Mobilenet Results

Chart, line chart

Description automatically generated

Figure 3: MobileNet First Training Sequence Accuracy and Loss

Chart

Description automatically generated

Figure 4: MobileNet First Training Sequence ROC AUC Score

Chart, line chart

Description automatically generated

Figure 5: Final Finetuned Mobilenet Accuracy and Loss

Chart, line chart

Description automatically generated

Figure 6: Final Finetuned MobileNet ROC AUC Score

Graphical user interface, text

Description automatically generated

Figure 7: Finetuned MobileNet Classification 1

Graphical user interface

Description automatically generated

Figure 8: Finetuned MobileNet Classification 2

Text

Description automatically generated

Figure 9: Final Finetuned MobileNet Classification 3

## Multi-Class Classification

Chart

Description automatically generated

Multi-Class Accuracy for Train and Validation

Chart, line chart

Description automatically generated

Multi-Class Loss for Train and Validation

Graphical user interface, application

Description automatically generatedA screenshot of a computer

Description automatically generated

Correct

Graphical user interface

Description automatically generatedA screenshot of a computer

Description automatically generated with medium confidence

Wrong

Graphical user interface, application

Description automatically generatedA screenshot of a computer

Description automatically generated with medium confidence

Wrong

## Siamese Network with Contrastive Loss Architectures

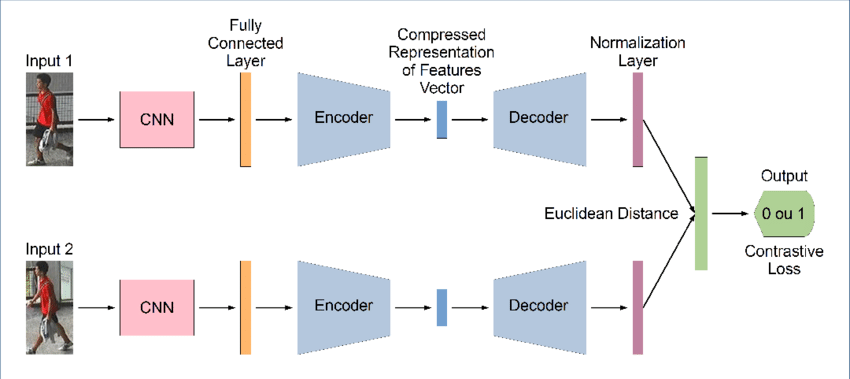


Figure 10: Siamese Network with Contrastive Loss & Euclidean Distance visual representation architecture

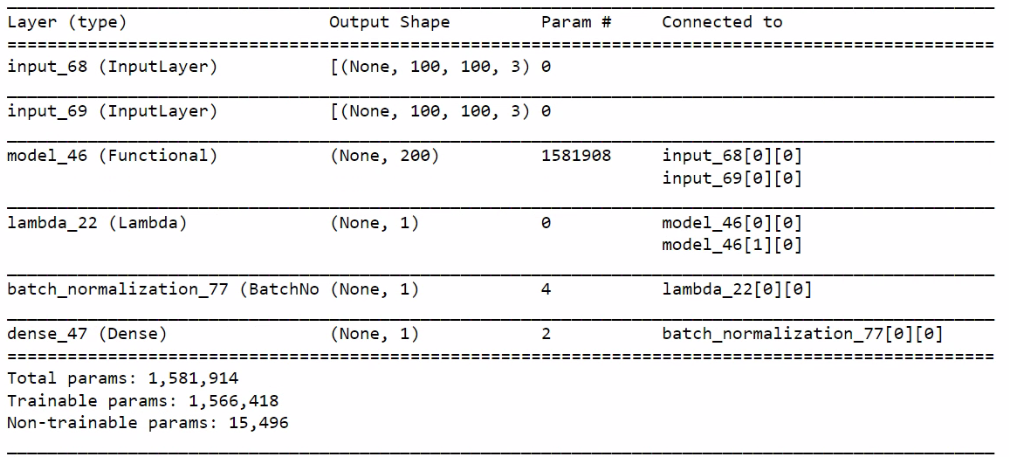


Figure 11: Siamese Networks with Embedding Networks 1 & 2 (The figure of each network combined as the architecture was similar)

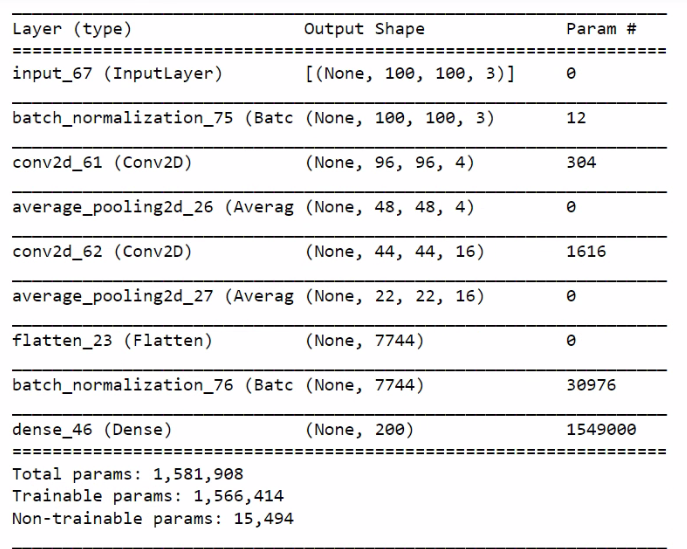
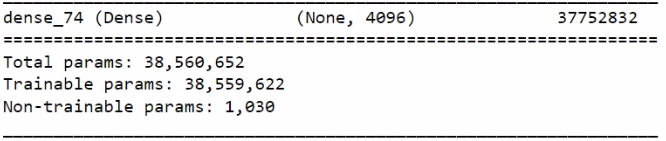
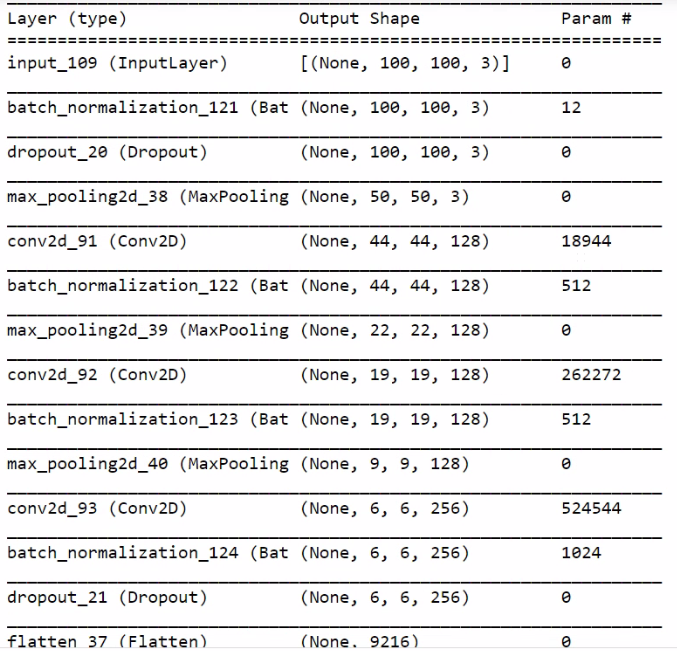


Figure 11: Embedding Networks 1 & 2 (The figure of each network combined as the architecture was similar)



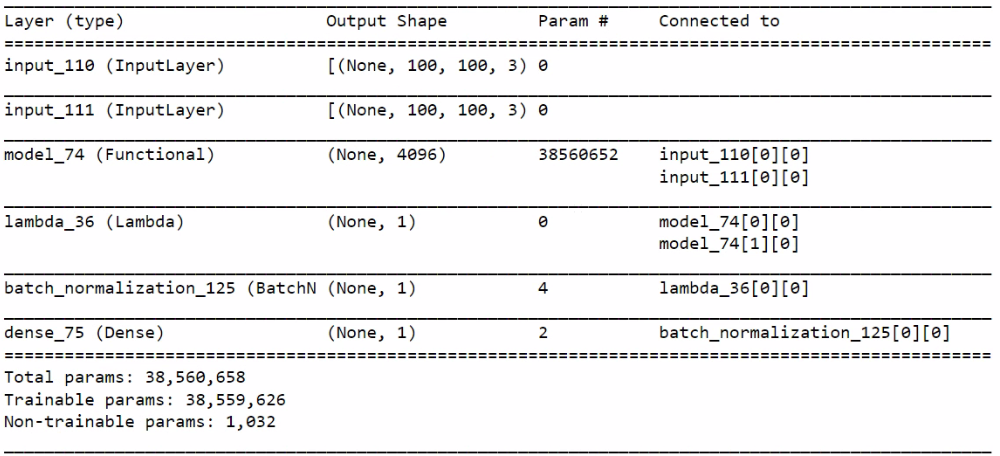


Figure 14: Siamese Networks with Embedding Network 3

Figure 13: Embedding Network 3

## Siamese Network Results

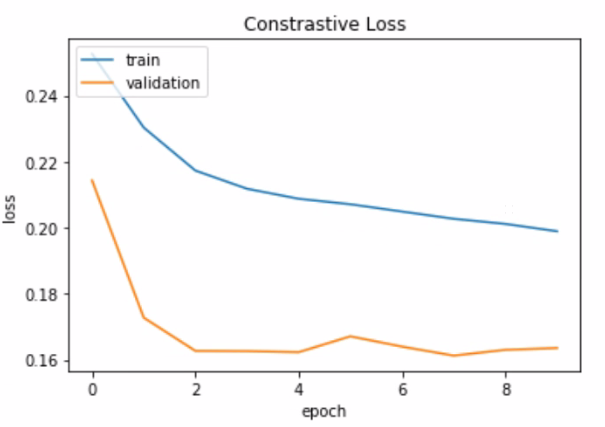
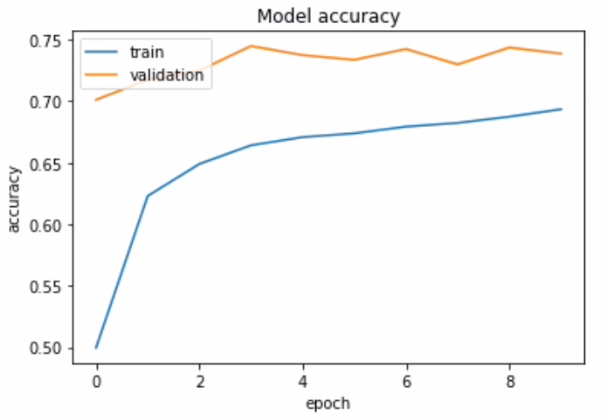


Figure 15: Siamese Network with Embedding Networks 1 Accuracy & Loss

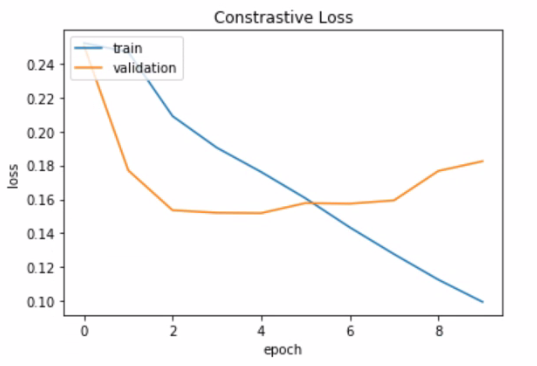
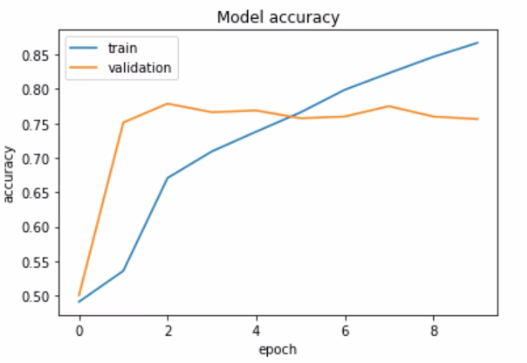
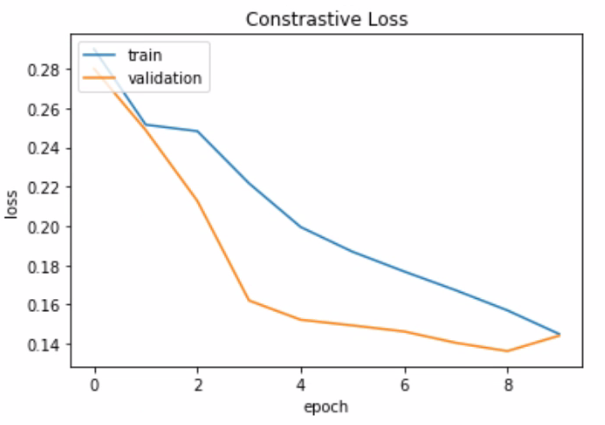
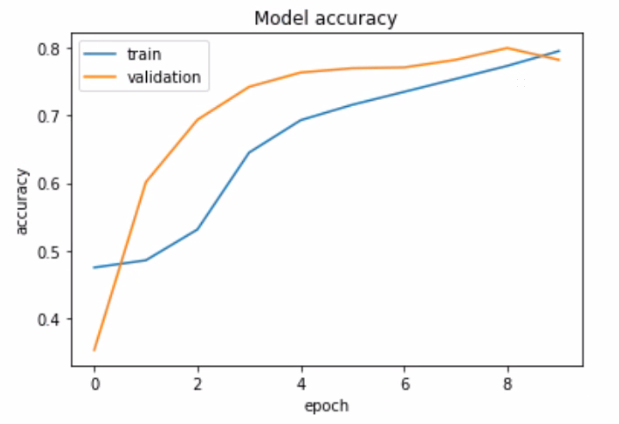


Figure 16: Siamese Network with Embedding Networks 2 Accuracy & Loss

Figure 17: Siamese Network with Embedding Networks 3 Accuracy & Loss



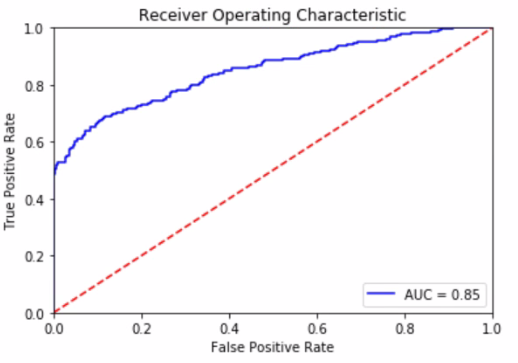
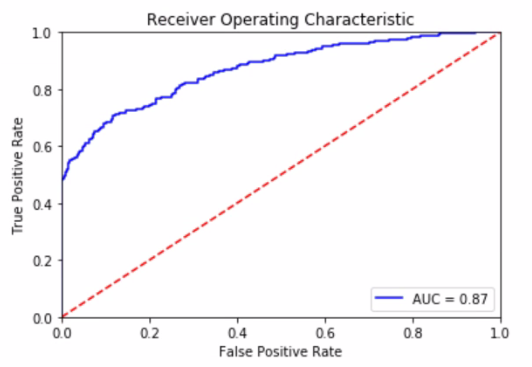
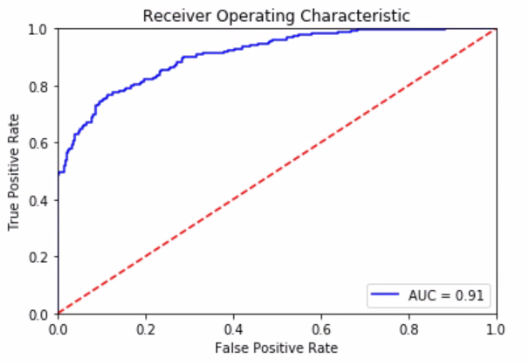


Figure 18: Siamese Network with Embedding Networks 1 ROC AUC Score

Figure 20: Siamese Network with Embedding Networks 3 ROC AUC Score

Figure 19: Siamese Network with Embedding Networks 2 ROC AUC Score



Figure 12: Siamese Network with Embedding Networks 3 Similarity. The threshold value is 0.5, if Pred value is >= 0.5 both images are considered as the same person. If Pred value is < 0.5 both images are considered as different person.

Text

Description automatically generated with low confidence

Figure 13: Attendance System Login

A screenshot of a video game

Description automatically generated with medium confidence

Figure 14: Face Verification Area after logged in

A screenshot of a video game

Description automatically generated with medium confidence

Figure 15: Unknown face detected

A screenshot of a video game

Description automatically generated with medium confidence

Figure 16: Face verified and detected with attendance taken

# 4 References

Bekuzarov*,* M 2020*, Losses explained: Contrastive Loss | by Maksym Bekuzarov | Medium*, Medium viewed 20 November 2021, <https://medium.com/@maksym.bekuzarov/losses-explained-contrastive-loss-f8f57fe32246>.

Koch, Zemel & Salakhutdinov n.d., *Siamese Neural Networks for One-shot Image Recognition*, CMU School of Computer Science viewed 22 November 2021, <https://www.cs.cmu.edu/~rsalakhu/papers/oneshot1.pdf>.

Nepal, P, 2020, *Mobilenet architecture explained*, Prabin Nepal, viewed 19 November 2021, < https://prabinnepal.com/mobilenet-architecture-explained/>.