Real-Time Face Recognition System

Shubham Jain, Gaurav Bajaj, Raj Chauhan, Raj Gariwala

***Abstract*— Face recognition is a biometric technology used for recognizing identity through the faces of humans. In this paper, a Real-time Face Recognition Model is developed using MobileNetV2 Convolution Neural Network. Data of 105 celebrities from Kaggle was used to train the MobileNetV2 model to classify images. In further developments in future, we will train it to generate embedding vectors for each human face of the celebrities. Later, these embedding vectors in the model (developed during model training) are used to identify the human faces from the live webcam.**

***Keywords***—  **MobileNet, Real-time, Face Recognition System, Deep Learning, Convolution Neural Networks, Computer Vision**

1. INTRODUCTION

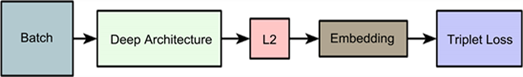
Face recognition is a method of recognizing an individual using their face. Face recognition systems can be used to identify individuals in photos, videos, or in real-time. The main motivation behind building a real-time face recognition system is to enable faster processing and seamless integration in various real-world applications such as public safety systems, authentication systems, retail-store systems, etc.

During the initial days, the speed of recognition of human faces in machines was slow and the accuracy seemed to be lower than manual recognition. However, with the development of deep learning and the increased application of Convolution Neural Network (CNN), the accuracy and speed of human-face recognition by machines are greatly improved.

We try to build a real-time face recognition system using a suitable model which is capable of detecting and recognizing images from webcams and cameras accurately. In real-time, we aim to develop the system such that it is capable of recognizing faces live, using a laptop webcam or cameras installed in a room.

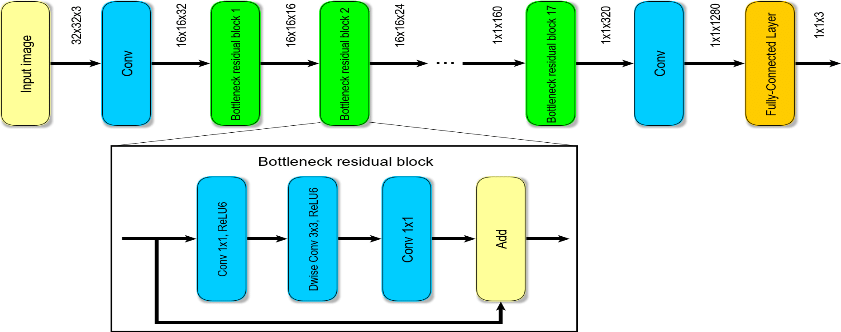
1. LITERATURE SURVEY

Deep neural network models, specifically CNNs, are greatly applied in computer vision tasks such as image classification and face detection and provide very accurate results. The problem of face recognition can be easily reduced to 2 sub-problems which are Face detection and Face recognition. Face detection is not just detecting the face but also cropping out unrelated parts in the image except the face.  
 FaceNet, a face recognition system developed by Google, uses deep neural networks to map images to Euclidean space. It is based on the concept that the distance between pictures of similar people in Euclidean space is little, and the distance between pictures of different people in Euclidean space is huge. FaceNet has provided over 99% accuracy in the Labelled Faces in the Wild. The FaceNet basic architecture includes 5 blocks - the input layers (batch), the deep convolutional architecture, L2 normalization, leading to face embedding data (face features), and lastly utilizing triplet loss to minimize the distance between similar faces.



**Fig. 1: Basic FaceNet Architecture**

However, the FaceNet model is complex with low calculation speed, more parameters, and multiple layers. Hence, ultimately affecting the overall size and speed of the face recognition system, especially when used in real-time.On the other hand, MobileNet architecture seems to be a good fit as it is a lightweight deep neural network, built upon deep separable convolutions layers and blocks. Its fast computation speed and low model size make it the most practical model for real-time face recognition. It consists of convolution layers with about 32 filters along with 19 residual bottleneck layers. Moreover, we can see that there are 7.5M parameters in FaceNet while only 3.4M in MobileNet V2.



**Fig. 2: MobileNetV2 Convolution Architecture**

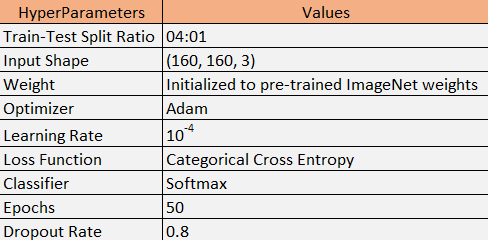
1. IMPLEMENTATION

For the project of real-time face recognition, we have used the TensorFlow Keras version of the MobileNetV2 model along with the 105\_classes\_pins dataset available over Kaggle. It is a dataset containing 17534 faces and images of about 105 celebrities. The best thing about this dataset is that it is well cropped and labeled which does not call for any large pre-processing. Till now, we have trained the MobileNetV2 model on the dataset and tested it upon the classification of static images.

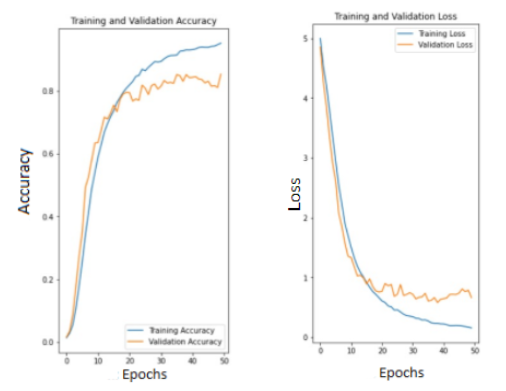
Due to less number of parameters which leads to less size and faster computation in the MobileNet model, it is considered great for real-time projects like that of face recognition where speed and space are crucial factors.

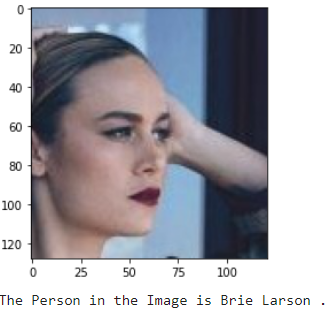
The dataset was split into testing and training set in the ratio 1:4 using python script with proper shuffling. For data augmentation, the ImageDataGenerator function of Keras library was used, which introduced random transformations like scaling, rotation, shearing, etc. in the batch of the training image.

MobileNetV2 model is trained using transfer learning with Imagenet weights used as pre-trained weights. All the pre-trained layers are unfrozen for re-training. Additional layers of Global Average Pooling, Dropout, and Dense (105) were also added to the model. The model was compiled with loss function as categorical cross-entropy and Adam (10-4) optimizer. The model is then run to fit the training data and test data over about 50 epochs. The statistics of training and testing are shown in the results section.

**Fig. 3: Hyperparameters**

IV. RESULTS

  
 **Fig. 4: Accuracy vs Epoch & Loss vs Epoch**



**Fig. 5: Successful recognition/classification of the static image by the model**

On model fitting, a training accuracy of 95% is reached with a loss of 18% in the final epoch. On the other hand, a validation accuracy of 85% is achieved with a loss of 62%.

We also tried to classify a test image by inputting it into the model. The model correctly classifies the image of the celebrity as is shown in the picture above.

V. CONCLUSIONS

As is observable from the results above, the model has a decent validation accuracy given the number of parameters in MobileNetV2. But at the same time, it can also be observed from the loss vs epoch graph that the model currently overfits the data. We are planning to use a larger dataset than what we have currently used or introduce dropout or reduce the learning rate to resolve the overfitting.

Further, our system currently is capable of recognizing static images rather than real-time faces from webcam or video. We aim to train the model based on facenet architecture, i.e., with triplet loss and then develop the project further for recognizing faces in real-time. Once our model is trained with triplet loss, we would store the 128D embeddings (feature vector) of the images in the dataset. When new faces are added, we would run those faces through the model and generate their embeddings (without re-training the model) and store it along with other stored embeddings. Embeddings of the faces detected in the camera (embeddings can be obtained through the trained model) will be compared for similarity against these stored embeddings and in this way, we are planning to recognize faces.

The constraints of real-time face recognition are described as follows: -  
(1) Illuminations: The images taken from real-time may have high or low illuminations or shadows. This may result in no or false face detection in real-time.  
(2) Distance: An important factor while detecting faces in real-time can be the distance of the camera from the face. If the distance is large, the human face might appear smaller in the camera.  
(3) Orientation: Orientation of the human face and the angle to the camera impacts the rate of face detection and recognition in real-time.  
(4) Face Occlusion: In some images, human faces may be partially hidden by objects like glasses, hats, hairs, scarves, masks, etc. This affects the detection and recognition rate of a face.

References

1. X. Xu, M. Du, H. Guo, J. Chang, and X. Zhao, “Lightweight FaceNet based on mobilenet,” *International Journal of Intelligence Science*, 01-Dec-2020. [Online]. Available: https://www.scirp.org/journal/paperinformation.aspx?paperid=104599. [Accessed: 20-Mar-2022].
2. “Face recognition,” *Electronic Frontier Foundation*, 15-Feb-2021. [Online]. Available: https://www.eff.org/pages/face-recognition. [Accessed: 20-Mar-2022].
3. “Why is facial recognition important - 5 key benefits - NEC New Zealand,” *NEC*, 26-Jan-2021. [Online]. Available: https://www.nec.co.nz/market-leadership/publications-media/why-is-facial-recognition-important-5-key-benefits/. [Accessed: 20-Mar-2022].
4. S. Ghosh, “How to create a face recognition model using FaceNet Keras?,” *Medium*, 10-Jul-2020. [Online]. Available: https://medium.com/clique-org/how-to-create-a-face-recognition-model-using-facenet-keras-fd65c0b092f1. [Accessed: 20-Mar-2022].
5. D. Kumar, “Introduction to facenet : A unified embedding for face recognition and clustering,” *Medium*, 21-Jun-2020. [Online]. Available: https://medium.com/analytics-vidhya/introduction-to-facenet-a-unified-embedding-for-face-recognition-and-clustering-dbdac8e6f02. [Accessed: 20-Mar-2022].
6. S.-H. Tsang, “Review: MOBILENETV2 - light weight model (image classification),” *Medium*, 01-Aug-2019. [Online]. Available: https://towardsdatascience.com/review-mobilenetv2-light-weight-model-image-classification-8febb490e61c#:~:text=MobileNetV2%20Overall%20Architecture&text=In%20typical%2C%20the%20primary%20network,and%20uses%203.4%20million%20parameters. [Accessed: 20-Mar-2022].
7. Mark Sandler, “MobileNetV2: Inverted Residuals and Linear Bottlenecks”, Cornell University, 21st March 2019. [Online] Available:https://arxiv.org/abs/1801.04381v4 [Accessed: 20-Mar-2022]