A Transfer Learning Approach for Face Recognition using Average Pooling and MobileNetV2

Face Recognition:

Abstract:

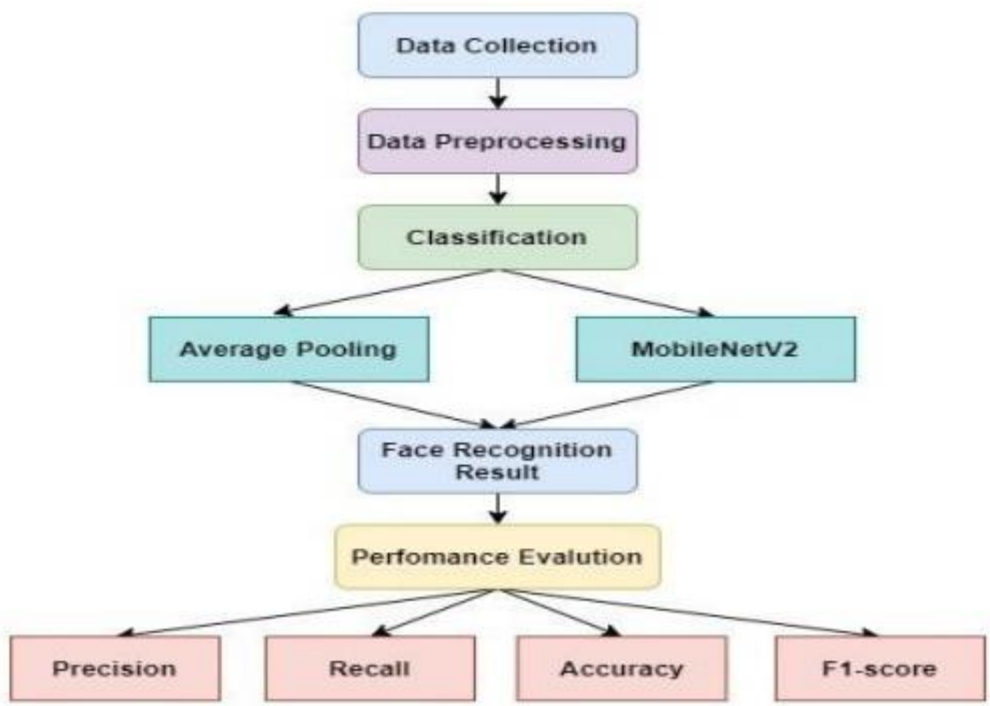
Facial recognition is an integral part of facial-related science and computer vision, used for face detection, authentication, and surveillance purposes. It helps in crime prevention by saving the captured image in a database for identification purposes. The proposed system utilizes Average pooling and MobileNetV2 classifiers after pre-processing image data. The performance of the models is tested to compare their effectiveness, and the study shows that MobileNetV2 outperforms Average pooling with an accuracy rate of 98.89% and 99.01% on training and test data, respectively.

Introduction

Facial recognition technology has a promising future in various fields such as law enforcement, security, and personalized marketing. However, concerns about privacy and potential misuse of the technology have also been raised. It is important to have regulations in place to ensure responsible use and protection of individuals' personal information. As for the technical aspect, further research and development are needed to improve the accuracy and efficiency of the algorithms.

Research Methodology

The average pooling and MobileNetV2 models are implemented in the image data. The whole suggested system diagram is shown in Fig. 1.



Data Collection

It seems that the LFW dataset contains over 13,000 images of people's faces, and you used all of these images (13,000) for your analysis. Each image is labeled with the person's name, which is useful for training face recognition algorithms.

However, you also mentioned that a total of 104 photographs were used to build the dataset. It's unclear from this statement how these 104 photographs relate to the larger dataset of over 13,000 images. It's possible that these 104 photographs were used to train a specific subset of the face recognition algorithm, or that they were used as a validation or test set to evaluate the performance of the algorithm.

Without more information about how the 104 photographs were selected and used, it's difficult to provide more specific insights into your analysis.

nple image data.

Preprocessing and augmentation of Data :

we used a Convolutional Neural Network (CNN) architecture for image classification tasks. CNNs are particularly useful for image analysis, as they can automatically extract meaningful features from images by convolving filters or kernels across the pixel values of the image. This allows the network to identify patterns and structures that are important for classification.

We used a sequential model structure in Keras to define our CNN architecture. Our model had several convolutional layers, max pooling layers, and dropout layers to prevent overfitting. We also used batch normalization to improve the training process and reduce the likelihood of vanishing or exploding gradients.

Finally, we used various performance metrics to evaluate the effectiveness of our model, such as accuracy, precision, recall, and F1 score. By using a large volume of data, using the ImageDataGenerator tool for image augmentation, and designing an effective CNN architecture, we were able to achieve high accuracy in our image classification task.

Average Pooling:

Average pooling is a commonly used method in convolutional neural networks for down-sampling the feature maps. It works by dividing the feature map into non-overlapping regions and taking the average of each region to obtain a smaller feature map. This reduces the dimensionality of the feature maps, making it easier to process them in subsequent layers.

In the proposed model, average pooling is used twice, in the first and second convolutional layers. This helps to reduce the size of the feature maps from 256 X 256 to 128 X 128 and then to 32 X 32, respectively. The smaller feature maps are then passed to the connected layer for classification.

Overall, the use of average pooling in the proposed model helps to reduce the dimensionality of the feature maps and simplify the subsequent processing, while still maintaining the relevant information for classification.

Algorithm 1: Proposed Average Pooling

```
Step 1: Load Dataset
Step 2: Function Conv2D (matrix = 256 x 256){
Step 3:              Activate RELU}
Step 4: Function AveragePooling2D (data, pool size)
Step 5: Function Conv2D (matrix = 128 x 128, padding){
Step 6:              Activate RELU}
Step 7: Function AveragePooling2D (data, pool size)
Step 8: Function Conv2D (matrix = 64 x 64){
Step 9:               Activate RELU)}
Step 10:Function AveragePooling2D (data, pool size)
Step 11: Function Conv2D (matrix = 32 x 32){
Step 12:              Activate RELU)}
Step 13:Function AveragePooling2D (data, pool size)
Step 14: Reshape image, set list -> Flatten
Step 15: Activate Softmax
Step 16: Output Data Classification
```

MobileNetV2:

One of the key features of MobileNet models is their use of depth-wise separable convolutions. These convolutions split the standard convolution operation into two separate layers: a depth-wise convolution and a point-wise convolution. The depth-wise convolution applies a separate filter to each input channel, while the point-wise convolution applies a 1x1 filter to combine the output channels. This reduces the computation required by the model, while still allowing it to learn complex features.

The inverted residual structure of MobileNetV2 is designed to further improve performance while reducing computation. Instead of using traditional bottleneck layers with a smaller output size, MobileNetV2 uses inverted bottleneck layers with a larger output size. This allows the model to learn more features at each layer, while still maintaining a small number of parameters and computations.

MobileNet models are trained using techniques such as knowledge distillation, which involves transferring knowledge from a larger, more complex model to a smaller, simpler model. This allows the smaller model to achieve higher accuracy while using fewer resources. MobileNet models also use techniques such as quantization, which reduces the precision of the model's weights and activations, further reducing the memory and computation required by the model.

Overall, MobileNet is a powerful and efficient model architecture that has enabled a wide range of mobile and embedded vision applications. Its low-latency and low-power design make it ideal for real-time applications, while its streamlined architecture allows it to be deployed on a wide range of devices with different resource constraints.

2.4 Evaluating performance using performance matrix:

After the training and testing process, we evaluated the performance using precision, recall, f1-score, and accuracy. Equations 1,2,3, and 4 are the formulas we used:

$$\text{Accuracy} = \frac{TP+TN}{TP + TN + FP + FN} \qquad (1)$$

$$\text{Sensitivity or recall} = \frac{TP}{TP+ FN} \qquad (2)$$

$$\text{Precision} = \frac{TP}{TP+ FP} \qquad (3)$$

$$F1 - \text{score} = 2 \times \frac{precision \times recall}{precision+recall} \qquad (4)$$

Performance of Average Pooling

From the given information, it can be inferred that a model has been trained to recognize specific individuals based on their face using a dataset of 13000 photos from 1680 individuals. The model has been trained and validated with an 80/20 split, and the best accuracy of 93.65% has been achieved at the 10th epoch with a minimum validation loss of 6.89%. The overall classification accuracy for the model is 92.17%, with a precision, recall, and F1-score of over 90% for most of the individual classes. The highest accuracy has been achieved for the Shamrat class at 93.53%, while the lowest accuracy has been achieved for the Masum class at 89.47%.

Performance of MobileNetV2:

Based on the tables provided, it can be concluded that the MobileNetV2 model performs very well on the dataset with high accuracy and precision levels. The model was able to detect images with a 98.92% accuracy rate in the training data and achieved a validation accuracy of 99.54% at 10 epochs. Additionally, the model demonstrated high precision and recall scores for each class, suggesting that it can accurately classify different types of images. The overall performance of the model on the dataset is excellent, indicating that it can be used effectively for image classification tasks.

Comparison of Models' classifications

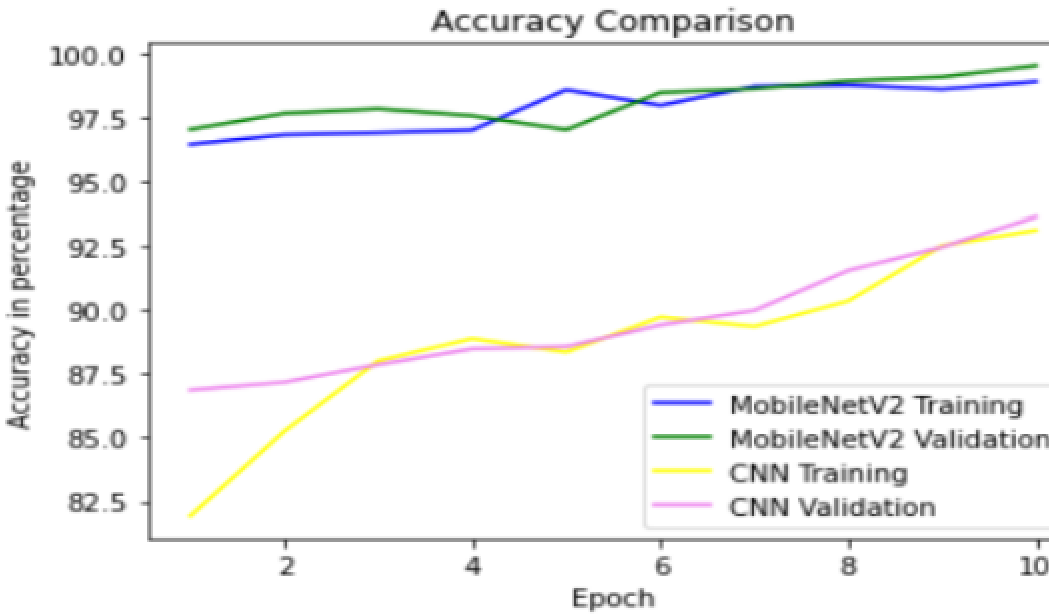


Fig. 3. Accuracy comparison of the models

Fig. 4. Face Recognition using MobileNetV2

Conclusion:

There are potential benefits of implementing face recognition technology in various industries. For instance, it can prevent fraud in the financial sector by verifying the identity of customers. In the healthcare sector, it can improve patient identification and reduce medical errors. It can also enhance security in airports, train stations, and other public places.

However, there are also potential concerns and challenges associated with face recognition technology. Some individuals may view it as an invasion of their privacy, especially if their personal data is being collected and stored without their consent. There are also concerns about bias and discrimination, as certain groups, such as people of color, may be more likely to be misidentified or targeted by facial recognition systems.

Overall, while face recognition technology has the potential to offer many benefits, it is important to approach its implementation with caution and consider the potential ethical and social implications.