IIT ROPAR
Department of Electrical Engineering

Single Image Low-Resolution Face Recognition

Prepared By:
Aena Ghai - 2020EEB1148
Drishti Jain- 2020EEB1168
Hiya Kwatra- 2020EEB1173
Muskan Singhvi- 2020EEB1295
Tejasvini Uppal- 2020EEB1032

Supervised By:
Dr. Jyotindra S. Sahambi

A Department Engineering Project submitted to the Department of Electrical Engineering in partial fulfillment of the requirements for the degree of B.Tech in Electrical Engineering

27th April, 2023

IIT ROPAR
Department of Electrical Engineering

Single Image Low-Resolution Face Recognition

Prepared By:
Aena Ghai - 2020EEB1148
Drishti Jain- 2020EEB1168
Hiya Kwatra- 2020EEB1173
Muskan Singhvi- 2020EEB1295
Tejasvini Uppal- 2020EEB1032

Supervised By:
Dr. Jyotindra S. Sahambi

A Department Engineering Project submitted to the Department of Electrical Engineering in partial fulfillment of the requirements for the degree of B.Tech in Electrical Engineering

27th April, 2023

JSSahambi
27.4.2023

Abstract

In this report, we are going to present a study on various models used for face recognition. We are going to dive deeper into their working, their advantages as well as limitations and explain how the quality of resolution affects the facial recognition technology.

Acknowledgement

We would like to extend our thanks and express our gratefulness to our supervisor, Dr. Jyotindra S. Sahambi and our course coordinator Dr. Mahendra Sakare for their necessary guidance and support throughout this project. Also, we would like to express our sincere gratitude to all those who supported us throughout the project, our friends and families, and we hope that our work will contribute significantly to the advancement of facial recognition technology and its responsible use in society.

Table of Contents

Abstract **II**

Acknowledgement **III**

Table of Contents **IV**

1 Motivation **1**

2 Objective **2**

2.1 Problem Statement 2

2.2 Problem Identification 2

2.3 Detailed description of problem 2

3 Literature Review **4**

3.1 Types of Resolution 4

3.1.1 Low resolution 4

3.1.2 High resolution 4

3.2 Use cases of HQFR and the need to shift on LQFR 5

3.3 How Low quality images are produced from high quality subjects 5

3.4 Current works in Face Recognition 6

3.4.1 Deep learning-based approaches 6

3.4.2 Super-resolution techniques 6

3.4.3 Transfer learning 6

3.4.4 GAN-based approaches 6

4 Dataset Description **7**

5 Methodology **8**

5.1 Approach 8

5.1.1 Inbuilt face_recognition library 8
5.1.2 GAN with FaceNet 10
5.1.3 Siamese Model 14

6 Results and Discussions **20**

7 Conclusion and Future Work **21**

8 References **22**

1. Motivation

Facial recognition technology has become increasingly popular in recent years, with applications ranging from security and surveillance to identity verification and access control. However, the accuracy of face recognition systems can vary greatly depending on the algorithm and model used. The motivation for this project is to evaluate the performance of various face recognition models and determine which performs best. By comparing different models, we can gain insights into their strengths and weaknesses, identify areas for improvement, and develop more accurate and reliable face recognition systems. Ultimately, this project aims to advance our understanding of face recognition technology and its potential use for real-world applications.

2. Objective

Contents

2.1 Problem Statement 2
2.2 Problem Identification 2
2.3 Detailed description of problem 2

2.1 Problem Statement

Nowadays, many diverse approaches are used for facial recognition. However, the issue in such systems arises when images of low resolution are used, images captured by surveillance cameras forms majority of it.

2.2 Problem Identification

Images obtained from surveillance cameras are degraded due to the the camera's software or hardware or due to the surveillance environmental conditions. Faces in these images can be indistinct due to defocus, subject or camera motion, or lack high frequency. In times when there is an increasing demand for surveillance, it is necessary to have a face recognition system which helps in analyzing faces/objects in images acquired by a surveillance camera.

2.3 Detailed description of problem

Today, there is an increasing demand for a high performance face recognition system due to globalization. Face recognition has attracted a massive attention over the past decades mainly due to its variety of applications. It has become popular in fields like image processing, security and medicine. Face recognition models consist of preprocessing of the images, feature extraction and recognition using various techniques. After extraction of features and processing, the face is compared to known faces present in a database. But images captured by surveillance cameras are degraded by low resolution, low contrast, blur, and noise. Models are affected during the face alignment stage as they are not trained to consider image distortions. So there is a need for low quality face recognition models. Hence, we will be working on low resolution degradation on high quality images i.e, training images by down-sampling them. The goal of resolving the problem is to improve the accuracy of low-quality face recognition models and strengthening security surveillance systems.

3. Literature Review

Contents

3.1 Types of Resolution 4
3.1.1 Low resolution 4
3.1.2 High resolution 4
3.2 Use cases of HQFR and the need to shift on LQFR 5
3.3 How Low quality images are produced from high quality subjects 5
3.4 Current works in Face Recognition 6
3.4.1 Deep learning-based approaches 6
3.4.2 Super-resolution techniques 6
3.4.3 Transfer learning 6
3.4.4 GAN-based approaches 6

3.1 Types of Resolution

3.1.1 Low resolution

Low resolution refers to images that have a low number of pixels per inch, which eventually results in a lower level of detail and clarity. This most probably makes the pixels in the images or videos discernable, blurry, or jagged. Low-Quality Face Recognition (LQFR) models performs poorly in the identification and verification of faces. That may be due to lower-resolution cameras used, less advanced algorithms , or not sufficient training data, which results in a higher number of false positives and false negatives. This leads to high probability that the face recognition might misidentify individuals.

3.1.2 High resolution

High resolution is referred to the display twhich has a high number of pixels per inch in comparison, resulting in a higher level of detail, clarity, and sharpness, making the image or video appear clearer, sharp and consisting of a better contrast. High-Quality Face Recognition (HQFR) systems are widely accepted and integrated in day to day systems today because of their efficient and accurate results. These systems comprise of high-resolution cameras, advanced algorithms, and are trained on large and complex datasets. These systems can be used in various applications like security, surveillance, mobile unlocking and making payments, etc.

3.2 Use cases of HQFR and the need to shift on LQFR

When high-quality photographs are available, such as passport photos, driver's license photos, or high-resolution images taken by professional photographers, HQFR (High-Quality Facial Recognition) systems are employed. For these systems to successfully identify a person, photographs must have high resolutions, excellent contrast, and good lighting. These days, automatic face recognition is beneficial and is referred to as the modern security entryway. The benefits of face recognition in surveillance cameras can include the following because our study focuses on these devices: Identification of criminals: The police and other civil agencies keep a database of criminals or those who are wanted. It is a success because the offender is identified if a detected face matches a face in a list of persons or if a face seen by one camera at one moment matches a face seen by another camera at the same time or at a different time. When low-quality photos are the only ones accessible and high-quality images are not, LQFR (Low-Quality Face Recognition) methods are employed. LQFR is the only option in a variety of circumstances, including dimly lit environments, Surveillance footage, and distantly taken pictures. Despite the poor image quality, LQFR algorithms may reasonably accurately identify a person's face in these situations.

3.3 How Low quality images are produced from high quality subjects

LQFR is important because in most cases images are captured in uncontrolled situations. Low quality images are produced when one or more than one of the following degradation processes are applied to high quality inputs:-

1. Low Resolution of capturing indices - The fewer the pixels, lower would be the resolution. Surveillance cameras are usually low quality cameras (as placing so many HD cameras would be quite expensive) and are always placed at an altitude. Hence low resolution images are produced due to low spatial resolution of the cameras and large camera standoff distances.

2. Blurriness - Surveillance cameras can produce blurry images due to a variety of reasons like the lens might be out of focus, the lens be dirty due to environmental factors, various hardware problems like the image sensors might be defective or connections might be loose or maybe the camera is not a high quality camera.

3. Acquisition Conditions - Inappropriate illumination conditions and pose lead to noise in images hence producing low quality images. Artifacts - Image artifacts can occur due to disturbances in hardware, image compression etc.

3.4 Current works in Face Recognition

High Resolution Facial Recognition (HRFR) has received the most attention to date, and the algorithms' outputs are practically flawless and highly accurate. To model HRFR, deep convolutional neural networks are employed. Deep CNN performs poorly on low quality images when compared to HRFR.There are also super-resolution-based techniques for LRFR, but they also need a lot of work to get better outcomes.

Low resolution face recognition is a challenging problem in computer vision because of the poor quality of the input images. Here are some current works in low resolution face recognition:

3.4.1 Deep learning-based approaches

With the increasing success of deep learning, many researchers have explored deep learning-based approaches for low resolution face recognition. These approaches use deep neural networks to extract features from low-resolution images and then use these features for recognition. Some of the recent works include "LRFR: A Deep Learning Approach for Low-Resolution Face Recognition" and "Low-Resolution Face Recognition via Learning Deep Representation."

3.4.2 Super-resolution techniques

Super-resolution models can be deployed to enhance the resolution of images, hence can improving the performance of face recognition systems. Recent works in this area include "Learning Face Super-Resolution Networks using Dynamic Convolution" and "Progressive Face Super-Resolution via Attention to Facial Landmark."

3.4.3 Transfer learning

Transfer learning is an approach where a pre-trained model is fine-tuned on a new dataset to improve the performance. Researchers have explored transfer learning-based approaches for low resolution face recognition, where pre-trained models are fine-tuned on low-resolution face datasets. Some of the recent works include "A transfer learning based approach for low-resolution face recognition" and "Low-Resolution Face Recognition using Transfer Learning from High-Resolution Models."

3.4.4 GAN-based approaches

Generative adversarial networks (GANs) have been used for various computer vision tasks, including low resolution face recognition. GANs can generate high-quality images from low-resolution inputs, which can improve the performance of face recognition systems. Recent works in this area include "Low-resolution face recognition via generative adversarial networks" and "Learning to Super-Resolve Faces with Facial Landmarks in Low-Resolution Images using GANs."

4. Dataset Description

Dataset used - CFPW: Celebrities in Frontal-Profile in the Wild Dataset Images depict celebrities in real-world, uncontrolled settings either directly facing the camera (frontal view) or with their head turned to the side (profile view), such as those found in social media posts, news articles, or private photographs. The varying lighting, stance, emotion, and occlusion that are frequently seen in such photographs make it difficult to identify superstars.

Generating our dataset:

3 Frontal images of 500 individuals were collected for generating the training dataset. 1 Frontal image of 500 individuals were collected for generating the testing dataset. Therefore, the total Images that we worked on are 2000. Before going forward with the face recognition, we performed some operations on the images(as per the requirements) to work on them. The operations include resizing the images, grayscaling them, converting them to an array of pixels and downsampling them(through bicubic downsampling) to get the low-resolution images.

Figure 4.1: Types of low-quality images used

5. Methodology

Contents

5.1 Approach **8**

5.1.1 Inbuilt face_recognition library 8

5.1.2 GAN with FaceNet 10

5.1.3 Siamese Model 14

5.1 Approach

This project aims to develop a single-image low-resolution face recognition model using the following different approaches. The following is a detailed description of the methodology used to develop the various models.

5.1.1 Inbuilt face_recognition library

In this model, we have used several inbuilt libraries, including face_recognition, PIL and cv2. Steps for implementation :

1. Preprocessing the images : The images are preprocessed by upsampling them to a higher resolution and converting them to RGB format. We use the cv2 library to convert the images to RGB format. A simple Sequential Model consisting of UpSampling layer is used.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 up_sampling2d (UpSampling2D  (None, 130, 130, 3)      0
 )

 up_sampling2d_1 (UpSampling  (None, 260, 260, 3)      0
 2D)

=================================================================
Total params: 0
Trainable params: 0
Non-trainable params: 0
_________________________________________________________________
```

Figure 5.1: Summary of model

2. Extracting face encodings : The face encodings for each image are extracted using the face_encodings module of face_recognition library. These encodings are essentially a numerical representation of the facial features that can be used to compare different faces. It returns a list of 128 dimensional numpy arrays corresponding to each face.

3. We stored the known face encodings and their corresponding labels in two separate lists. The labels represent the identity of each face in the known set.

4. Recognition of unknown faces : We then compare the unknown face encodings with the known face encodings stored in the list to recognize the unknown faces and make the prediction label list. We use the compare_faces function in face_recognition library to compare the face encodings and retrieve the closest match.

Results for different test data are as follows:-

1. Upsampled Images Used : Accuracy Score - 0. 926

2. Downsampled Images Used : Accuracy Score - 0.922

3. Gaussian Blur applied to Downsampled Images : Accuracy Score - 0.478

Overall, the methodology involved preprocessing the images, extracting facial features, building the model, recognizing unknown faces, and visualizing the results. This approach allowed us to accurately recognize faces and match them with their corresponding identities.

Python's inbuilt fac_recognition library had the following disadvantages:

1. Face recognition with Python's inbuilt face_recognition library can be speed inefficient on large and complex datasets.

2. Model Accuracy: Although the built-in library provides a decent level of precision, it might not be useful for some tasks that need for extremely high accuracy rates. By using and combining different models and alternative data preprocessing techniques, we can increase accuracy.

3. Model Scalability: The built-in library might not be able to effectively handle the increased complexity and quantity of faces as the dataset gets bigger.

5.1.2 GAN with FaceNet

GAN with Keras FaceNet: The GAN with Keras FaceNet was specifically designed for face recognition. It is able to learn highly discriminative features that are invariant to variations in pose and lighting, and is computationally efficient enough to be used in real-time applications. The approach involves using a Generative Adversarial Network (GAN) to generate images which contain s high-resolution face from low-resolution inputs, followed by using a pre-trained Keras FaceNet to extract facial features from the generated images.

GAN Model:

GANs are used for the implementation of Single Image Super Resolution by keeping the quality under check and restoring the images. In the machine learning limbo there are two main classes of models, generator model and discriminator model.

1. In the GAN model, we have a Generator which generates fake images and a discriminator which declares whether an image is natural or generated by generator. Hence if the discriminator marks an image correctly, this implies that the generator model needs to improve its working otherwise the discriminator model needs to improve its working. Hence the generator and discriminator are trained mutually.
2. Images are extracted from the dataset (Initial Size : 384X384) and downsampled (bicubic downsampling) the new size is 96X96. The image is further normalized with all pixel values ranging from [-1,1].
3. Generator Model : The Generator model uses various Conv2D and Upsampling layers to generate upsampled images.

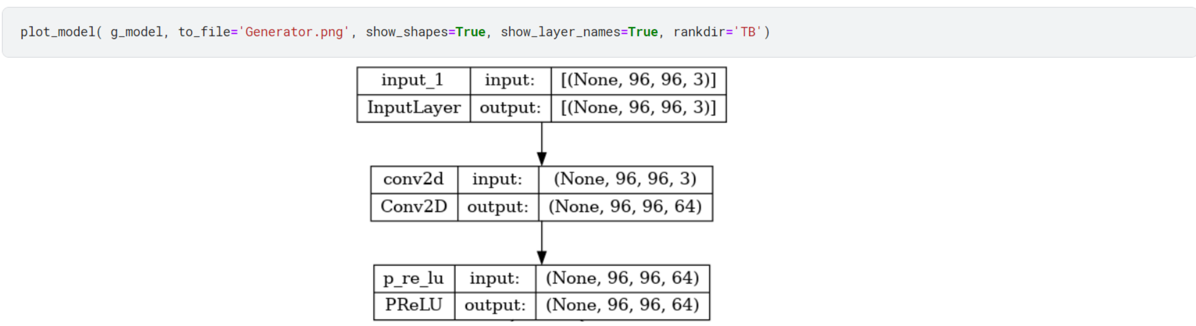

Figure 5.2: Generator model

4. Discriminator Model : Discriminator model use various 2D Conv Layer.

```
plot_model( d_model, to_file='Discriminator.png', show_shapes=True, show_layer_names=True, rankdir='TB')
```

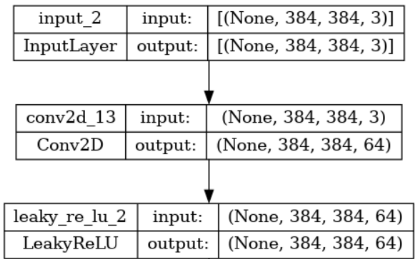

Figure 5.3: Discriminator model

5. Finally, the complete GAN model is made by combining the Generator and Discriminator using Adam optimizer (learning rate=0.0001, beta_10.9 , beta_2, epsilon = 1e-8). The model is trained using 200 epochs and a batch size of 64 images randomly selected from the total data.

That being said, there are several disadvantages of using GAN:

1. We faced the issue of the "Memory Limit being Exceeded" when the GAN model was run on GPU (Kaggle Compiler), due to which the results that could have been obtained by GAN model were not beneficial.

2. GAN models require a massive amount of dataset, highly efficient processors and time to train and generate realistic high resolution image images.

3. Also, GAN models can be computationally expensive to train, which put a contraint on money and time.

The three very popular algorithms that can be combined together for face recognition tasks: MTCNN, FaceNET and SVM Face recognition system can be divided into three categories:

STEP 1: Face detection -> STEP 2: Feature Extraction-> STEP 3: Feature Matching

Face Detection: The method used: MTCNN

Multi-Task Cascaded Convolutional Neural Network is a neural network algorithm that finds the face window/frame boundary on the images and then extracts facial features. It is basically a face detection algorithm that detects and locates faces in images. It can detect differently scaled faces and differently oriented faces and uses a cascade neural network to perform the following tasks on the images containing faces:

1. Face Classification takes the aid of cross-entropy function for loss calculation which calculates the difference between the predicted and the true/actual probability distribution of the target class labels.

2. Bounding Box Regression: uses Euclidean for Loss function.

3. Facial Landmark Location: this uses the Euclidean loss function too and marks the points of the face for the localisation of the facial landmarks.

It can even work for faces that are oriented and is one of the most accurate and precise in terms of getting the bounds/window/frame of a face, however, the only drawback is that it's slower. The output of the MTCNN detector:

```
image = cv2.imread(OriginalImages[0])
image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
bboxes = detect_face(image)
print("Output of MTCNN detector is...\n",bboxes)
```

```
1/1 [==============================] - 0s 352ms/step
1/1 [==============================] - 0s 129ms/step
1/1 [==============================] - 0s 33ms/step
1/1 [==============================] - 0s 28ms/step
1/1 [==============================] - 0s 29ms/step
1/1 [==============================] - 0s 29ms/step
1/1 [==============================] - 0s 27ms/step
1/1 [==============================] - 0s 25ms/step
1/1 [==============================] - 0s 204ms/step
WARNING:tensorflow:5 out of the last 520 calls to <function Model.make_predict_function.<locals>.pred
1/1 [==============================] - 0s 175ms/step
Output of MTCNN detector is...
 [{'box': [37, 40, 167, 219], 'confidence': 0.9999855756759644, 'keypoints': {'left_eye': (86, 122),
```

Figure 5.4: MTCNN detector output

We use the box parameters specifically for our problem statement since our region of interest is the pink box as in the above picture, i.e. the face.


```
image = cv2.imread(OriginalImages[0])
image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
bboxes = detect_face(image)
print("Output of MTCNN detector is...\n",bboxes)
```

```
1/1 [==============================] - 0s 352ms/step
1/1 [==============================] - 0s 129ms/step
1/1 [==============================] - 0s 33ms/step
1/1 [==============================] - 0s 28ms/step
1/1 [==============================] - 0s 29ms/step
1/1 [==============================] - 0s 29ms/step
1/1 [==============================] - 0s 27ms/step
1/1 [==============================] - 0s 25ms/step
1/1 [==============================] - 0s 204ms/step
WARNING:tensorflow:5 out of the last 520 calls to <function Model.make_predict_function.<locals>.pred
1/1 [==============================] - 0s 175ms/step
Output of MTCNN detector is...
 [{'box': [37, 40, 167, 219], 'confidence': 0.9999855756759644, 'keypoints': {'left_eye': (86, 122),
```


Figure 5.5: Face Detection

Feature Extraction:

FACENET Utilising the Keras framework, deep learning face recognition algorithm of FaceNet is built. A shared convolutional base network, a global pooling layer, and a fully connected layer that creates the final embedding vector make up the three primary/main parts of the network design. A series of convolutional layers are followed by max pooling layers in the shared convolutional base network, which is utilised to extract features from the input facial image. The global pooling layer reduces/lowers the dimensionality/dimensions of the feature maps produced by the convolutional base network. The pooling method utilised is called global average pooling, and it averages(total/number) each feature map over its spatial dimensions. As a result, each feature map produces a single vector, which is then summed up together to create the final feature embedding.

The final embedding vector (embedded_x) is generated by the fully connected (FC) layer using the concatenated feature embedding as input. For the embedding vectors to have unit length, a normalisation layer like L2 normalisation is frequently added after this layer, which normally contains fewer neurons than the convolutional base network.
It offers a pre-trained FaceNet model that provides an excellent face embeddings vector and dense vector representations of facial features, locations and pounts and descriptions that are applied to face recognition tasks.

FaceNet model has its input as a face image(which we extracted from MTCNN, cropped according to the window/frame bounds and gave as an input to FaceNet) to produce a vector of 128/512 integers representing the most crucial facial characteristics which we called and is actually known as an embedding.
These embeddings are then further used in the training (SVM, here) classification of the dataset. It is trained with the triplet loss for the loss calculation, which promotes the vectors for the same identity to become more similar (basically minimized anchor-positive distance), i.e. have a smaller distance, and the ones with different identities to get lesser similarity or have a larger distance (basically minimized anchor-negative distance).

Feature Matching:

SVM

SVMs or Support Vector Machines classify the classes of the training data according to the different features of the face by creating an optimal hyperplane for the same. The embeddings vector is the one we extracted in the FaceNet part of our model. The hyperplane has dimensions one lower than the features one. SVMs learnt a hyperplane that separates the feature vectors (in this case, face embeddings) of different individuals in the space of the feature space. The hyperplane is selected/chosen such that it maximizes the margin between the two classes(the 2 differet ones), making it more robust/strong/sturdy to noise and outliers in the data.

The FaceNet model is used to extract embeddings from face images, which are then used as input to an SVM classifier. Now, during the training, the SVM distinguishes/differentiates between the embeddings of different individuals and during testing, it predicts the identity of a new face by classifying its embedding using the learnt decision boundaries from the previous learnigns.

Results for different test data from CFPW datasets are as follows:-

1. Original Images from CFPW dataset used : Accuracy Score - 0.9734/97.34
2. Upsampled Images Used : Accuracy Score - 0.6953/ 69.53

5.1.3 Siamese Model

Theory

In computer vision applications like face recognition and feature learning, deep metric learning has been a critical component in learning more discriminative features. Its main goal is to promote interclass differences and minimising intraclass variance. To gauge the degree of correspondence between the properties of Siamese pairings, we set out to train a Euclidean distance metric. We want the distance measure to have as large of a significant value towards negative instances and as little of a significant value towards positive examples. Siamese networks are those neural networks sharing the weights between its two sister networks, each of which generates their unique embedding vector on the basis of its own input. These train using either triplet loss or contrastive loss, which is more appropriate for the models and tends to increase accuracy.

Dataset Manipulation

From the CPFW-dataset used, first we used ' cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)' function to grayscale the image. Then we reshaped the images to 256x256 pixels for uniformity. to convert the dataset into low resolution images collection, we implemented bi-cubic downsampling to scale all the images to 65x65 pixel ratio.

Our Model Explained

In our work, we have suggest a Siamese network-based face recognition framework model. This strategy seeks to build a distance metric tailored on deep features of visual pairs generated by the Siamese network rather than simple up-sample and image similarity models. The model was trained to distinguish between pictures of celebrities from their created classes. Celebrity 1 must be distinguished from the other celebrities (1–500), for instance. So for celebrity 1, we chose N random photographs from class A and paired them with N random images from class B for celebrity 2. Then, until reaching 500, we continued this procedure for all classes. After matching up Celebrity 1, we did the same for the remaining classes for the remaining celebrities (numbering from 1 to 500).The Siamese network design consists of 2 fully linked layers and 5 convolutional layers. A contrastive loss function is used to train the network and penalises the model for correctly identifying similarities between photos of distinct people and vice versa. There are two input layers, each of which connecting to a different network and generates a different set of their own embeddings. The final network then uses the combined output to estimate the differences on a scale of 1 after Lambda layer blends them all together using a defined Euclidean distance function. Using the Visualise Function, we displayed our findings. It visualizes image pairs and their corresponding labels and predictions using Matplotlib. Predictions are used in the case of testing and contains the predicted output corresponding to the image pairs. It also highlights predictions regarding dissimilarity.

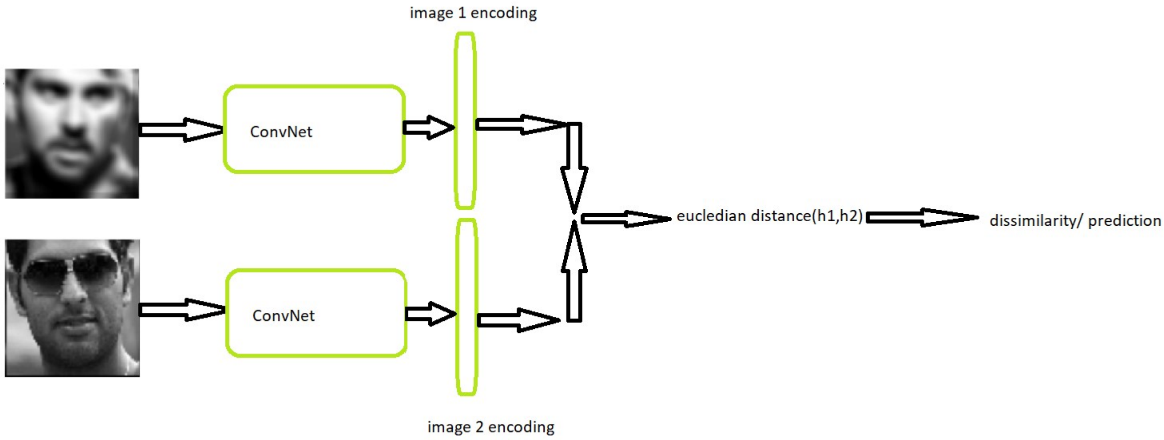

Figure 5.6: Siamese illustration

```
Model: "model_1"
__________________________________________________________________________________________________
 Layer (type)                   Output Shape         Param #     Connected to
==================================================================================================
 input_2 (InputLayer)           [(None, 65, 65, 1)]  0           []

 input_3 (InputLayer)           [(None, 65, 65, 1)]  0           []

 model (Functional)             (None, 10)           39590       ['input_2[0][0]',
                                                                  'input_3[0][0]']

 lambda (Lambda)                (None, 1)            0           ['model[0][0]',
                                                                  'model[1][0]']

 batch_normalization_2 (BatchNo  (None, 1)           4           ['lambda[0][0]']
 rmalization)

 dense_1 (Dense)                (None, 1)            2           ['batch_normalization_2[0][0]']

==================================================================================================
Total params: 39,596
Trainable params: 34,184
Non-trainable params: 5,412
__________________________________________________________________________________________________
```

Figure 5.7: Model Summary

Graphs for Accuracy and Contrastive Loss

Given are the graphs for trained and test data on model accuracy and contrastive loss with increasing epochs.

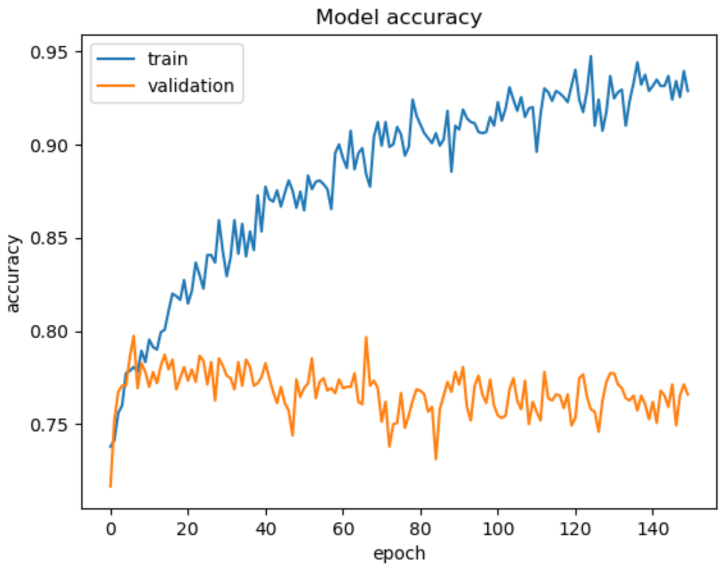

Figure 5.8: Model Summary: Model Accuracy

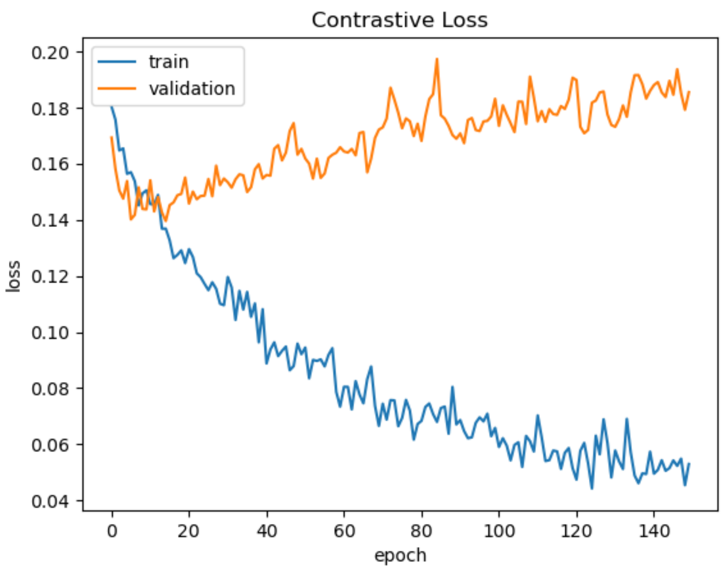

Figure 5.9: Model Summary: Contrastive Loss

Test cases

1. **On Gaussian Blur**
   We selected a downsampled image from our Downsampled Images array. Using 'Gaussian blur' function in cv2, we added blur to the already downsampled image to further degrade its quality. Then we passed both the low-quality images through our Siamese network to find the prediction of dissimilarity between the two.
   The prediction came out to be 0.00001, which signifies 0.001% dissimilarity. We can safely conclude that both the faces in the images are similar, and the face is recognized with a high probability.

Figure 5.10: Comparison with Gaussian Blur

2. **On Different Images of the Same Person**
   We selected two different downsampled images from our Downsampled Images array of the same person. Then we passed both the low-quality images through our Siamese network to find the prediction of dissimilarity between the two.
   The prediction came out to be 0.00051, which signifies 0.051% dissimilarity. We can safely conclude that both the faces in the images are similar and the face is recognized with a high probability.

Figure 5.11: Different Images Of Same Person

3. **On Females**
   We selected two different downsampled images from our Downsampled Images array of the different females. Then we passed both the low-quality images through our Siamese network to find the prediction of dissimilarity between the two.
   The prediction came out to be 0.99601, which signifies 99.601% dissimilarity. We can safely conclude that both the faces in the images are dissimilar, with a high probability.

Figure 5.12: Same Gender Compared

4. **On Different Genders**
   We selected two different downsampled images from our Downsampled Images array of the different genders. Then we passed both the low-quality images through our Siamese network to find the prediction of dissimilarity between the two.
   The prediction came out to be 0.99577, which signifies 99.577% dissimilarity. We can safely conclude that both the faces in the images are dissimilar, with a high probability.

Figure 5.13: Different Genders Compared

5. **On Different Races/Skin Color**
   We selected two different downsampled images from our Downsampled Images array of people of different skin color/race. Then we passed both the low-quality images through our Siamese network to find the prediction of dissimilarity between the two.
   The prediction came out to be 0.99601, which signifies 99.601% dissimilarity. We can safely conclude that both the faces in the images are dissimilar, with a high probability.

Figure 5.14: Different Races Compared

Advantages of Siamese Network for Face Recognition

Siamese networks have an edge over various other models for face recognition.

1. One-shot learning algorithm: With just one image per individual, our network can distinguish faces in a dataset that has very few samples. This is useful in situations where there are many persons in the image who need to be identified but it is not practical to collect that many samples per person.

2. Model's robustness:Siamese networks are excellent for face recognition in real-world situations where image quality is very low because the lighting and poses of the face in the image can vary significantly. This is because siamese networks learn to extract features invariant to such variations.

3. Metric learning schematic: Siamese networks operate on the premise that they must first learn the degree of similarity between various pairs of images before they can compare and finally recognise faces.As a result, they can acquire incredibly diverse and intricate associations between faces and eventually deal with situations in which faces may be similar but not exactly the same.

4. Model's scalability: Siamese networks are ideal for real-world applications since they scale up quickly to handle large datasets and can be trained on several GPUs. This allows us to solve the problem at hand..

6. Results and Discussions

In this project, we developed three approaches to build a single-image low-resolution face recognition model: an inbuilt model, GAN with FaceNET and Siamese model. The performance of the developed models was evaluated using the accuracy metric on a test set.

The accuracy of the inbulit face_recognition model was 47.8%. However, the library model does not prove to be scalable and is not be able to handle the increased complexity of large datasets.

The GAN approach proves to be ideal for generating upsampled images. However, GANs are very computationally expensive to train, especially for large-scale image datasets. When upsampled images were tested on the FaceNET model, an accuracy of 69.53%, was achieved.

The Siamese model approach achieved an accuracy of 84.5%. The Siamese model outperformed the FaceNET approach in terms of accuracy. The Siamese model's ability to compare facial features in two images and determine if they belong to the same person led to a higher accuracy rate. On the other hand, the FaceNET approach generated high-resolution images but failed to extract the relevant facial features from the generated images, resulting in lower accuracy.

The results of this project are consistent with previous studies that have shown the effectiveness of Siamese models in face recognition tasks. Siamese models have the advantage of being able to compare the features of two images directly, making them more effective in recognizing faces even in low-resolution images.

However, it is worth noting that the GAN model has potential in improving the quality of low-resolution images before feeding them to the face recognition model. By generating high-resolution images, the GAN approach could help the Siamese model extract more accurate and relevant facial features from the images, potentially improving the overall accuracy of the model.

The results of this project demonstrate the potential of deep learning techniques in low-resolution face recognition tasks. The developed models can be used for various applications, including surveillance, access control, and identity verification. Future research could explore the use of other deep learning techniques such as transfer learning and attention mechanisms to further improve the accuracy of the developed models.

7. Conclusion and Future Work

Our studies highlighted how deep learning and machine learning techniques could help in implementing low resolution face recognition systems with higher accuracy than the conventional or traditional low resolution recognition systems. The models implemented in this project can be used in various fields and domain that includes surveillance and verification of identity. At last, the Siamese model approach seemed most scalable and efficient in improving the accuracy of low resolution face recognition systems.

Future work could explore merging the GAN model with FaceNET approach to improve the quality of images before feeding them to the Siamese model. Additionally, other deep learning techniques such as transfer learning and attention mechanisms can also be used along these models to improve the efficieny of face recognition systems.

Overall, this project contributes in the area of improving face identification and recognition of low-resolution images and provides a much greater scope for research in this area. Improvement in the accuracy and efficiency of the low-resolution face recognition systems help in the improvement of the security and effectiveness of various systems that depends on face recognition technology.

8. References

[1] Heidari, Mohsen & Fouladi, Kazim. (2020). Using Siamese Networks with Transfer Learning for Face Recognition on Small-Samples Datasets. 1-4. 10.1109/MVIP49855.2020.9116915.

[2] L. S. Luevano, L. Chang, H. Méndez-Vázquez, Y. Martínez-Díaz and M. González-Mendoza, "A Study on the Performance of Unconstrained Very Low Resolution Face Recognition: Analyzing Current Trends and New Research Directions," in IEEE Access, vol. 9, pp. 75470-75493, 2021, doi: 10.1109/ACCESS.2021.3080712.

[3] A. M. Putra, A. Zaini and E. Pramunanto, "Implementation of MTCNN Facial Feature Extraction on Sleepiness Scale Classification Using CNN," 2022 International Conference on Computer Engineering, Network, and Intelligent Multimedia (CENIM), Surabaya, Indonesia, 2022, pp. 1-8, doi: 10.1109/CENIM56801.2022.10037269.

[4] Zhixue Wang, Chaoyong Peng, Yu Zhang, Nan Wang, Lin Luo. "Fully convolutional siamese networks based change detection for optical aerial images with focal contrastive loss", Neurocomputing, 2021