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

This report presents a study on face recognition using various models. The study involves a thorough review of the literature on GAN model, Siamese Network, and other face recognition techniques. The results obtained from the experiments demonstrate the effectiveness of the proposed approaches. This study provides valuable insights into the use of face recognition and also seeks to explain how facial recognition technology is affected by low and high resolutions.

Acknowledgement

We would like to express our sincere gratitude to everyone who supported us throughout the development of this face recognition project. Firstly, we extend our heartfelt thanks to our supervisor, Dr. Jyotindra S. Sahambi and our course coordinator Dr. Mahendra Sakare for their valuable guidance and support throughout the project. Their encouragement, expertise, and insights were invaluable in shaping our ideas and approach. We would also like to thank the faculty and staff of the Electrical Engineering Department for their support and resources. Their dedication to education and research has provided us with the tools and knowledge to undertake this project. We are also grateful to our fellow students, who have provided us with invaluable feedback, support, and encouragement throughout the project. Their contributions have helped us to refine our ideas, identify areas for improvement, and stay motivated throughout the process. Finally, we would like to thank our families and friends, who have provided us with unwavering support, love, and encouragement. Their belief in our abilities has been a constant source of motivation, and we could not have completed this project without them. Once again, we express our sincere gratitude to all those who have supported us throughout the project, and we hope that our work will contribute to the advancement of facial recognition technology and its responsible use in society.

Table of Contents

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

In recent times, diverse biometric approaches have been the foundation of an increasing number of automatic access systems. Face recognition systems are characterized by low acquisition invasiveness and are highly intrusive as they rely on capturing, extracting, storing, or sharing people's biometric facial data. However, the issue that arises in such systems is when the low resolution of images are used. Some major challenges to the performance of face recognition systems arise from low-resolution images captured by surveillance cameras.

2.2 Problem Identification

Typically, images obtained from surveillance cameras are degraded due to the circuitry, limitations of 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 to integrate security with surveillance, it becomes 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

In this modern age, there is an increasing demand for a high performance face recognition system due to globalization. Face recognition has attracted a great deal of attraction over the past decades mainly due to its variety of applications and useful image analysis. It has become popular with time In fields like image processing, security and medicine. Face recognition models primarily consist of preprocessing of the images, feature extraction and recognition using classification techniques. After extraction of features and processing, the face is compared to known faces present in a database.The existing facial recognition models fail in terms of performance when used in surveillance systems as compared to when used in relatively well-controlled environments. This is because images captured by surveillance cameras are degraded by low resolution, low contrast, blur, and noise. Unconstrained low-resolution face recognition, compared to many face recognition models tested on high-resolution images captured in controlled environments, is often very poorly explored. Models are particularly affected during the face alignment stage as they are not trained to consider image distortions. Surveillance security cameras provide low quality images due to low cost and limited storage space. So there is a need for low quality face recognition models. Owing to the lack of appropriate data sets of real-life surveillance situations, in this work we will be simulating low resolution degradation on high quality images i.e, training images by down-sampling them. Maintaining the cost of cameras, to code and deploy this application on various on field tasks. The goal of resolving the problem is to keep surveillance camera costs low while improving the accuracy of low-quality face recognition models, thereby strengthening security surveillance systems and allowing this application to be deployed to various on-field tasks.

3. Literature Review

Contents

3.1 Types of Resolution

3.1.1 Low resolution

Low resolution refers to an image, video, or display that has a low number of pixels per inch, resulting in a lower level of detail and clarity. This can make the image or video pixelated, blurry, or jagged. Low-Quality Face Recognition (LQFR) refers to systems that are less accurate in identifying and verifying faces. These systems may have lower-resolution cameras, less advanced algorithms, or insufficient training data, which can result in a higher number of false positives and false negatives. This means the system may misidentify individuals or fail to recognize them altogether.

3.1.2 High resolution

High resolution, on the other hand, refers to an image, video, or display that has a high number of pixels per inch, resulting in a higher level of detail, clarity, and sharpness. This makes the image or video appear clearer and more vibrant. High-Quality Face Recognition (HQFR) refers to highly accurate and reliable systems which identify and verify faces. These systems often have high-resolution cameras, advanced algorithms, and are trained on large amounts of diverse data. This means that the system can identify individuals accurately and can be used in various applications, from security and surveillance to unlocking mobile devices or making payments.

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:- Low Resolution - 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. 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. 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 success of deep learning in various computer vision tasks, 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 techniques can be used to enhance the resolution of low-resolution face images, which can improve 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 a popular approach in deep learning 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." Overall, low resolution face recognition is an active area of research, and researchers are constantly exploring new techniques to improve the performance of face recognition systems on low-resolution images.

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

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.

While Python's inbuilt fac_recognition library is a great tool for face recognition, there are a few disadvantages:

1. Performance: Face recognition with Python's inbuilt face_recognition library can be slow, especially when processing large datasets with multiple faces.

2. Accuracy: While the inbuilt library offers a good level of accuracy, it may not be sufficient for some projects that require very high accuracy rates. We can improve the accuracy by using other methods for preprocessing data and various models.

3. Scalability: As the size of the dataset grows, the inbuilt library may not be able to handle the increased complexity and number of faces efficiently.

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 high-resolution face images 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 high resolution images from low resolution 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.

```
plot_model( g_model, to_file='Generator.png', show_shapes=True, show_layer_names=True, rankdir='TB')
```

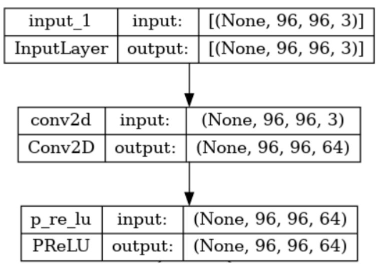

Figure 5.2: Generator model

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')
```

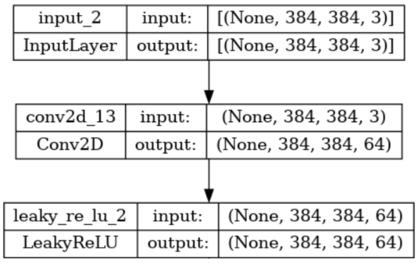

Figure 5.3: Discriminator model

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 potential disadvantages of using GAN:

1. Major Issue:- 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 obtained by GAN could not be used.

Other Issues that might occur due to the usage of GANs are:-

1. Limited training data: GANs require a large amount of training data in order to generate realistic images. For face recognition, there may not be enough high-quality data available to train a GAN effectively.
2. High computational cost: GANs can be very computationally expensive to train, especially for large-scale image datasets like those used for face recognition. This can make training a GAN for face recognition prohibitively expensive.

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:

Face detection->Feature Extraction->Feature Matching

Face Detection: The method used: MTCNN

MTCNN, or Multi-Task Cascaded Convolutional Neural Network, is a neural network which detects faces and facial landmarks on images.
It is basically a face detection algorithm that detects and locates faces in images. It can detect faces of different scales and orientations and uses a cascade neural network to perform the following tasks:

It performs the following three tasks:

1. Face Classification: uses cross-entropy loss function to binary classify Cross entropy: It measures the difference between the predicted probability distribution and the true probability distribution of the target class labels.
2. Bounding Box Regression: uses Euclidean Loss function.
3. Facial Landmark Location: this, too, uses the Euclidean loss function and marks for the localisation of the facial landmarks.

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

```
Output of MTCNN detector is...
 [{'box': [0, 0, 219, 264], 'confidence': 0.9916648268699646, 'keypoints': {'left_eye': (57, 90), 'right_eye': (159, 88), 'nose': (102, 151), 'mouth_left': (65, 202), 'mouth_right': (148, 200)}}]
1/1 [==============================] - 0s 113ms/step
1/1 [==============================] - 0s 112ms/step
1/1 [==============================] - 0s 38ms/step
1/1 [==============================] - 0s 39ms/step
1/1 [==============================] - 0s 34ms/step
1/1 [==============================] - 0s 35ms/step
1/1 [==============================] - 0s 36ms/step
1/1 [==============================] - 0s 38ms/step
1/1 [==============================] - 0s 145ms/step
1/1 [==============================] - 0s 158ms/step
```

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

This deep learning face recognition model makes use of the Keras framework. The network architecture consists of three main components: a shared convolutional base network, a global pooling layer, and a fully connected layer that generates the final embedding vector. The shared convolutional base network, used to extract features from the input facial image, comprises a succession of convolutional layers followed by max pooling layers. The dimensionality of the feature maps generated by the convolutional base network is lowered using the global pooling layer. Global average pooling, which calculates the average of each feature map along its spatial dimensions, is the pooling technique used. This produces a single vector for each feature map, which is concatenated to form the final feature embedding.

The final embedding vector is generated by the fully connected 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 can provide excellent face embeddings and dense vector representations of facial features that may be applied to face recognition tasks.

FaceNet uses a face image as its input and produces a vector of 128 integers that represent the most crucial facial characteristics. This vector is known as an embedding.
These embeddings are then further used in the training classification of the dataset. It is trained with the triplet loss function, which promotes the vectors for the same identity to become more similar, i.e. have a smaller distance, and the ones with different identities to get lesser similarity or have a larger distance.
The goal is to minimize the distance between the anchor and positive embeddings while maximizing the distance between the anchor and negative embeddings.

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 hyperplane has dimensions one less than the features. SVMs work by learning a hyperplane that separates the feature vectors (in this case, face embeddings) of different individuals in the feature space. The hyperplane is chosen such that it maximizes the margin between the two classes, making it more robust 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. During training, the SVM learns to distinguish between the embeddings of different individuals, and during testing, it can predict the identity of a new face by classifying its embedding using the learned decision boundary.

SVMs have several advantages for face recognition, including their ability to handle high-dimensional feature spaces and their robustness to noise and outliers in the data. However, they can also be computationally expensive to train and may not perform as well as more advanced deep-learning models in certain scenarios.

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

1. Original Images from CFPW dataset used : Accuracy Score - 0.9734

2. Upsampled Images Used : Accuracy Score - 0.6953

5.1.3 Siamese Model

Theory

Deep metric learning has been a critical element in learning more discriminative features in computer vision tasks, such as face recognition and feature learning. Its core idea is to encourage reducing intra-class variance and enlarging interclass differences. We aimed to train a distance metric (Euclidean distance) to measure the extent of correspondence between features of Siamese pairs. We want the distance metric to get a significant value as much as possible toward negative examples and a value as small as possible towards positive examples. Siamese Networks are neural networks sharing weights between two sister networks, each producing embedding vectors of their respective inputs. These use either contrastive or triplet loss when training which is better suited for the models and tends to improve 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 propose a face recognition framework based on the Siamese network. Instead of basic up-sampling and image similarity models, this approach aims to learn a distance metric customized on deep features of visual pairs generated by the Siamese network. We trained the model to differentiate between images of celebrities of different classes. For example, celebrity 1 needs to be differentiated from the other celebrities (1 through 500). We selected N random images from class A (for example, for celebrity 1) and paired them with N random images from another class B (for example, for celebrity 2). Then, we repeated this process for all classes of digits (until celebrity 500). Once we have paired Celebrity 1 with others, we repeated this process for the remaining classes for the rest of the celebrities (from 1 to 500).The Siamese network architecture has 5 convolutional layers and 2 fully connected layers. The network is trained using a contrastive loss function, which penalizes the model for predicting that two images of different persons are the same and vice versa. There are two input layers, each of them leading to its own network, producing its own embeddings. Then Lambda layer merges them both using a Euclidean distance function defined, and the merged output is fed to the final network to predict the differences on a scale of 1. We visualized our results using Visualize Function. 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.

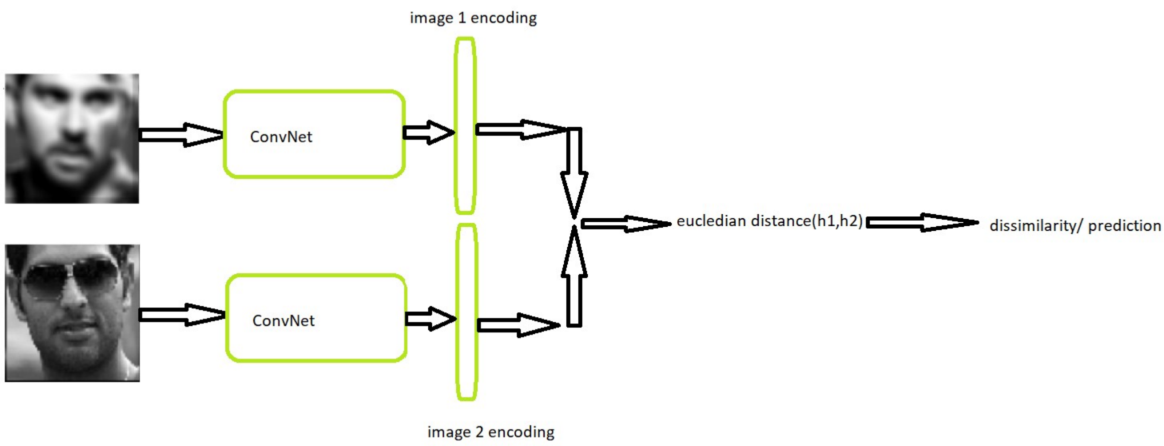

Figure 5.6: Siamese illustration

```
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.

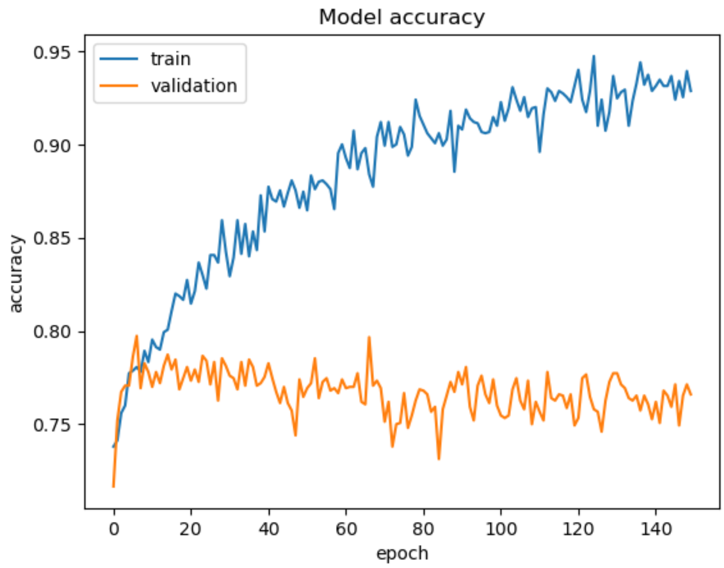

Figure 5.8: Model Summary

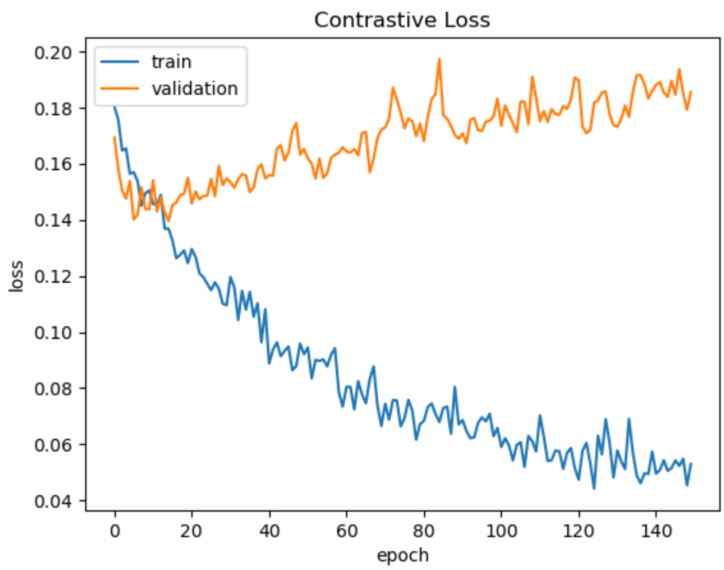

Figure 5.9: Model Summary

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: Same Gender 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 several benefits over other models for face recognition, including:

1. One-shot learning: Siamese networks can be trained to recognize faces with very few samples, even as few as one image per person. This is especially useful in scenarios where there are a large number of people to recognize and it is not practical to collect a large number of samples for each person.

2. Robustness to variations: Siamese networks are robust to variations in lighting, pose, and facial expression, as they learn to extract features that are invariant to such variations. This makes them ideal for face recognition in real-world scenarios where lighting conditions and poses can vary significantly.

3. Metric learning: Siamese networks learn a similarity metric between pairs of images, which can be used to compare and recognize faces. This allows them to learn complex relationships between faces and to handle cases where faces may be similar but not identical.

4. Scalability: Siamese networks can be easily scaled to handle large datasets and can be trained on multiple GPUs, making them suitable for real-world applications.

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

In this project, we developed three approaches to build a single-image low-resolution face recognition model: inbuilt face recognition model, GAN with FaceNET and Siamese model. Our experiments demonstrated that the Siamese model approach outperformed the FaceNET approach in terms of accuracy for low-resolution face recognition. 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.

Our study highlights 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. The Siamese model approach has the potential to be incorporated into various systems that require low-resolution face recognition, improving their accuracy and efficiency.

Future work could explore incorporating the GAN with FaceNET approach to improve the quality of low-resolution images before feeding them to the Siamese model. Additionally, other deep learning techniques such as transfer learning and attention mechanisms could be explored to further improve the accuracy of the developed models.

Overall, this project contributes to the field of low-resolution face recognition and provides a basis for future research in this area. By improving the accuracy and efficiency of low-resolution face recognition systems, we can improve the security and effectiveness of various systems that rely 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