Identifying Lost Children using Deep Learning

B.E. Project Report

Submitted in partial fulfillment of the requirements

For the degree of

Bachelor of Engineering
in
Computer Engineering

Submitted by

Ms. Amisha Mangesh Pupala 18CE5004
Ms. Samruddhi Manohar Mokal 18CE5007
Ms. Neha Ananta Pandit 18CE5022

Guided by

(Mrs. Smita Bharne)

Department of Computer Engineering

Ramrao Adik Institute Of Technology

Dr. D. Y. Patil Vidyanagar, Sector-7, Nerul, Navi Mumbai-400706.

(Affiliated to University of Mumbai)

December 2020

Ramrao Adik Institute of Technology

(Affiliated to the University of Mumbai)

Dr. D. Y. Patil Vidyanagar, Sector-7, Nerul, Navi Mumbai-400706.

CERTIFICATE

This is to certify that, the project 'A' titled

" Identifying Lost Children Using Deep Learning"

is a bonafide work done by

Ms. Amisha Mangesh Pupala 18CE5004
Ms. Samruddhi Manohar Mokal 18CE5007
Ms. Neha Ananta Pandit 18CE5022

and is submitted in the partial fulfillment of the requirement for the degree of

Bachelor of Engineering
in
Computer Engineering
to the
University of Mumbai

Supervisor
(Mrs. Smita Bharne)

Project Co-ordinator	Head of Department	Principal
(Mrs. Smita Bharne)	**(Dr. Leena Ragha)**	**(Dr. Mukesh D. Patil)**

Project Report Approval for B.E

This is to certify that the project 'A' entitled ***" Identifying Lost Children Using Deep Learning "*** is a bonafide work done by ***Ms. Amisha Mangesh Pupala***, ***Ms. Samruddhi Manohar Mokal***, and ***Ms. Neha Ananta Pandit*** under the supervision of ***Mr. Smita Bhare***. This dissertation has been approved for the award of ***Bachelor's Degree in Computer Engineering, University of Mumbai***.

Examiners :

1. ....

2. ....

Supervisors :

1. ....

2. ....

Principal :

....

Date : 16/12/2020

Place : Nerul, Navi Mumbai

Declaration

We declare that this written submission represents my ideas in my own words and where other's ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Ms. Amisha Mangesh Pupala	**18CE5004**	____________________
Ms. Samruddhi Manohar Mokal	**18CE5007**	____________________
Ms. Neha Ananta Pandit	**18CE5022**	____________________

Date : 16/12/2020

Abstract

A countless number of children goes missing, many cases left unsolved, problems in the identification of these missing children is causing issues which are becoming serious day by day and causing various types of problems to the officials and mainly for the parents of these missing children. Our system proposes a solution using deep learning, in which it gives a virtual space where the photograph of the missing child will be given at the time of reporting of his/her parents will get stored and the provision of adding photographs when any suspicious child get spotted will be given to the public so voluntarily they can take part in it. So the face recognition using the aging feature is displayed in our system. The input of the pre-processing step is the raw facial image, and the output is an aligned frontal face image. The objective of this step is to register all images, i.e., based on eye coordinates, such that they are aligned on the same standard size, i.e., each point on any given face is aligned to the same point of all of the images. The next step of the proposed system uses the FaceNet model to extract the face features. And this FaceNet model is an application of the very deep CNN architecture that is trained especially on a very large scale face database. whenever the suspicious child gets spotted. The public can upload photos of the suspicious child with details like place, time, landmarks, and remarks. This image will go through the Age C-gan algorithm. This model will be used for the synthesis of human faces within required age categories. This age processed image is given to the Facenet algorithm for feature extraction and face recognition. Learning the mapping from the given images and also creating the embedding rather using any type of layer for the task of recognition and verification is the major difference between the FaceNet and the other techniques. Which gives more and more accuracy in the results.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

When it comes to the safety of children we have to think thoroughly about them, as Children are the important factor of our society after all they are the future which we look up to, but in India a countless number of children missing cases get reported every year. According to the 2019-report of Hindustan Times" On average, a child goes missing every 10 minutes in India". And the main problem is that many of the cases are unsolved. A missing child from one region could be found in any another region but the difficulty occurs in the recognition of the child. Means even if a child get found it becomes difficult to recognize him/her from the reported missing cases which includes the photograph. But this arising issues which becomes the barrier in the identification of the missing child can be solved. The proposed idea is completely based on the maintaining the virtual space, such that at the time of the reporting photograph (recent photographs) of the missing child which is given by the parents is stored in a repository. The provision will be given to the public to voluntarily take the photographs of the children so they can add the photograph to see if there any match for it. To find the photograph among the missing child cases automatic searching will take place. So, it will match the photo using the aging feature to overcome the issues arises in the identification due to the aging of the children and it will be very beneficial to identify any suspected children even after the 5 or 10 years of their missing report. All barriers will get clear so the identification process will go smoothly and child will get found quickly, this will be very useful for the police officers to locate the missing child anywhere in India. And to keep our India's future safe.

1.1 Overview

As children's are the important aspect of our country, but due to cases of missing children's arising it is becoming like a threat to the future of our country. And finding any missing child with one photograph is a difficult job for the officials, as recognition plays a vital role in solving the case of missing children. And when the barriers come in the part of the recognition it becomes very difficult to solve the case and find the children who is missing, which can take 5 t0 10 years and that's why many children's are interested till date which is becoming the main reason into the increasing cases of the missing children. Sometimes it takes 5 to 10 years to solve case so the aging also becomes barrier in the recognition and then it becomes more difficult to recognize the children from the photograph which was given at the time of reporting, So the aging also plays an important role in the recognition and then it becomes more difficult to recognize the children from the photograph which was given at the time of reporting, So the aging also plays an important role in the recognition. Our system overcomes all these barriers which create difficulties in the recognition part. Using the deep learning features like FaceNet algorithm for recognizing any children using the aging feature (C-gan) can solve the issues and the process can go smoothly which will be very helpful for solving the cases.

1.2 Objective

Recognizing a missing child is a difficult job for the police and also for the higher authority, and even if a child is found, it becomes difficult to recognize him/her from the reported missing cases which include the photograph. Recognizing or identifying a child using face detection will make the process much easier. The proposed idea is based on the maintaining database, such that at the time of the reporting photograph (recent photographs) of the missing child which is given by the parents will get stored and the provision will be given to the public to voluntarily take the photographs of the children so they can add the photograph to the portal, so the automatic searching will take place. So, the objective of this project is to help Police and higher authorities track down missing children quickly.

1.3 Motivation

- Children are an important factor in our society, after all, they are the future which we look up to.
- But many children are get reported missing every year, which creates a big issue as most of them are interested.
- This problem arises because of the difficulties that come with the identification of the missing child. It is needed to solve this problem so these missing children can have a better future and to reduce the number of cases that are increasing day by day.

1.4 Organization of report

The report is organized by considering the topics as Literature Survey, Proposed Work, Results and Analysis, Conclusion and Future Works. Where the literature survey is consisting of the survey we have gone through, for example, the papers that we have used for our survey which have helped us to understand the condition and also the design of our system. Whereas the actual work and the methodologies are the algorithms we have used for our system is consisting of the topics as proposed work which have the details of the algorithm with the working of it and the software used for the implementation, fulfilled with the whole implementation details. The actual working means how the system will actually work in real life and implementation of the system is shown in the result and analysis with the actual pictures (snapshots) of the system which has been implemented. And the last is the conclusion of the project ”Lost Children Identification using Deep Learning” with the future work which is given in detail that we can do further improvement in the system and the implementation which we can carry out in future so it will be more beneficial.

Chapter 2

Literature Survey

2.1 Survey of Existing System

NAME OF THE PAPER	NAME Of THE TECHNOLOGY	INTRODUCTION	ADVANTAGES	DISADVANTAGES
1) Missing Child Identification System Using Deep Learning and Multiclass SVM	CONVOLUTIONAL NEURAL NETWORKS (CNN) VGG-FACE CNN DESCRIPTOR	CNNs are essential tools for deep learning methods and are more appropriate for working with image data. CNNs or ConvNets are composed of series of interconnected layers and these layers consist of repeated blocks of convolutional, ReLU pooling layers and fully connected layers. VGG-Face network is used for face recognition.	Performance these technologies is tested using the photographs of children with different lighting conditions, noises and also images at different ages of children. The Classification achieved a higher accuracy of 99.41	This technologies involves complex algorithms which make the process of extraction and classification slower.

2) Double-Blinded Finder: A Two-Side Privacy-Preserving Approach for Finding Missing Children	MULTITASK FACENET MODEL AND IPE METHOD FOR BLIND FACE MATCHING	Face-to-face matching can be safely run on the public cloud due to the face matching method based on IPE-based blind computing to restrict access to photos	Using these technologies, the system can achieve practical performance of blind face matching with negligible privacy leakage of both the suspicious and true missing children sides	solutions, especially internal products with threshold encryption schemes, should be improved to accommodate more effective face recognition applications in a secure environment
3) Child Face Recognition with Deep Learning.	VGG16, RESNET50, MOBILE-FACENET	In this paper, three Convolutional Neural Networks (VGG16,ResNet50 and Mobile-FaceNet) were proposed. MobileFaceNet is a lightweight model compared to other two models	Among three proposed CNN methods Mobile-FaceNet gives higher accuracy than others and processing time is faster than other two.	VGG16 requires largest model size and MobileFaceNet is a smallest model size.
4)Optimizing deep neural network structure for face Recognition	DEEPID MODULES WITH WIDE MODULE.	In this paper, DeepId, FaceNet networks are adjusted by adopting modified Inception structures which have different branches.	The mutual information of a single image in the same layer of different designed networks grows at the beginning as the width increases. However, the mutual information decrease with additional branches in wide module.	A deep face network shows the best performance when the difference between wide module and deep module is nearly equal

Table 1: Literature survey of Existing System

- Paper[1] uses Convolution Neural Networks and VGG Face descriptors for Face Recognition. In this paper, the author mentioned that CNNs are essential tools for deep learning methods and are more appropriate for working with image data. CNN or Conv-Nets are composed of series of interconnected layers and these layers consist of repeated blocks of convolutional, ReLU (rectified linear units), pooling layers, and fully connected layers.

These learned features were used to train a multi-class SVM classifier. They used this method to correctly identify and label the kid.

- Paper[2] applies IPE-based blind computing to restrict access to Images on Multitask FaceNet Model. It used the MT-FaceNet model to describe a child's face as a 128d fixed-point feature vector as well as auxiliary attributes. In this system, Face-to-face matching can be safely run on the public cloud due to the face matching method based on IPE-based blind computing. Using these technologies, the system can achieve a practical performance of blind face matching with negligible privacy leakage of both the suspicious and true missing children sides.
- Paper[3] uses three Convolutional Neural Networks (VGG16, ResNet50, and MobileFaceNet) were proposed. MobileFaceNet is a lightweight model compared to the other two models. It works efficiently and fast on embedded devices because of its depth-wise separable convolutional layers. Among the three proposed CNN methods, MobileFaceNet gives higher accuracy than others and the processing time is faster than the other two.
- Paper[4] uses DeepId, FaceNet networks are adjusted by adopting modified Inception structures which have different branches. Among these, A deep face network shows the best performance when the difference between the wide module and deep module is nearly equal.

2.2 Limitations of Existing System

- Existing systems do not contain an Age-progression module that can recognize images of the child having an age gap larger than 5 years or more.
- The existing system does not give high accuracy while recognizing the face in an uncontrolled environment like low light, blur images, images are taken from a distance, changes of a pose, illumination, expression, etc.
- The system uses the technologies which involve complex algorithm which make the process of extraction and classification of images slower.

2.3 Problem Statement

India is one of the countries that control a fifty fifth of the population of youth and kids. And this fifty-fifth is taken into account because the vital facet of our country, however, several ranges of kids are getting rumored missing each year that creates a giant issue as most of them are interested. This downside arises as a result of the difficulties that come with the identification of the missing kid. However, this downside may be solved using a system that is predicated on the deep learning methodology for the identifying of the reported missing child from the photos with face recognition. Because the missing kid may be found in any region, thus with the landmarks and remarks public will upload the photograph of the suspicious kid into the common portal. Automatic comparison can present itself with the photo that the public has posted and therefore the registered photo of the missing kid from the repository. A deep learning model is trained as there is going to be a method of classification for the input photo of the kid and therefore the photo that is matching best can get selected from the information of the missing children. To help the authorities in the cases of the missing children which will be beneficial for our society.

2.4 Scope

Missing children cases are arising day by day because of the problem in the identification, and this problem arises because it is difficult to find a child with a photograph with hundred percent surety and when the time goes by like after 5 to 0 years the problem of age gap creates more difficulties in the identification and after 5 or 10 years later identifying a child with an old photograph is much harder than we think. And police officers and other higher authorities have to go through these difficulties and more importantly the missing child and his/her parents have to face all these difficulties. So, by observing such conditions our system will not only help in the identification but even after 5 or 10 years later if any suspected child gets found the system will recognize the child correctly with the feature of an age gap and deep learning. Therefore, it will be much easier for the police officer and the parents of the missing child to identify the missing child quickly and easily, it will help the process go smoothly and perfectly.

Chapter 3

Project Proposal

3.1 Proposed Work

It is known that the performance of the face recognition algorithm is affected due to aging. It is still a problem to consider the aging parameter under consideration for face recognition as most of the algorithm fails to detect the face of the child after a few years.

A child's face undergoes various changes including facial hair, weight, the texture of the skin, shape, and size of the face, etc. Several studies were performed to analyze the extent to which facial aging affects the performance of face recognition algorithms. Two conclusions can e drawn from these studies:

1. The performance of traditional face recognition algorithms decreases with an increase in time-lapse.

2. Performance of face recognition and detection algorithms decreases more rapidly in the case of younger individuals as compared to older individuals.

Hence it is important to consider the age progression of the face for enhancing the performance of the face recognition algorithm, especially when enrolled at a young age.

In particular, our contributions are as follows:

- A portal for finding missing children that compare the image of the missing child with the images of the already registered missing children.

- To take into consideration many kinds of variations in face images which are taken under uncontrolled conditions such as change of expression,, illumination, pose, change in lighting condition, presence of noise and blurred image.
- Using Age Conditional Generative Adversarial Network to reconstruct high-quality synthetic images of the required age group preserving the original person's identity.

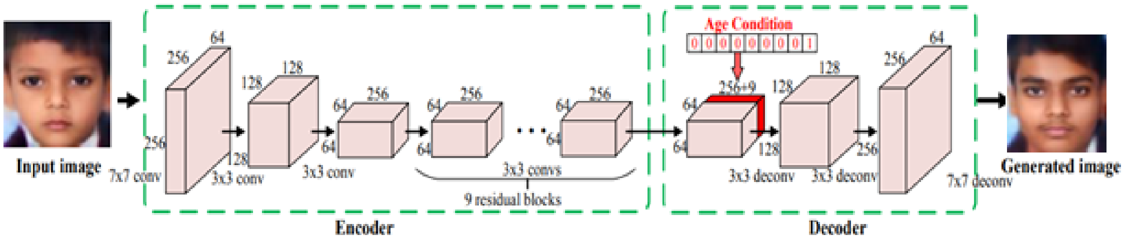


Figure 1: Layers used in Aging cGAN

Here we propose a methodology for missing child identification and recognition that combines facial feature extraction and age progression based on deep. Thus the proposed system utilizes face recognition for missing child identification. This is to help authorities and parents in missing child investigation.

Thus the proposed system presents a novel deep learning methodology to find the missing children use the photo of the children currently available using face recognition. Whenever anyone finds any suspicious child on the road he/she can click the photograph of the child and upload it into the common portal with landmarks and remarks. The public can upload the photograph of the missing child into the portal. Then the photo will automatically b compared with the already registered images in the repository. If no matching record is found then the image will be age progressed and again the age-progressed image will be compared with the registered photos. For this, a deep learning model Aging C-GAN is used. And the face recognition is done using FaceNet. Thus the proposed system outperforms the earlier methods in face recognition based on missing child identification.

As mentioned earlier face aging model is based on an age-conditional generative adversarial network, a model for synthesis and generation of human faces of the required age category. Once Age-cGAN is trained, the face aging is done in two steps

1. Given an input face image x of age y0, find an optimal latent vector z which is used to generate a reconstructed face x¯ = G(z , y0) as close as possible to the initial image.

2. Given the target age ytarget, generate the resulting face image xtarget = G(z, ytarget) by simply switching the age at the input of the generator.

Thus the task of our cGAN network is to generate the image of a child at different ages based on the input image and target age. The structure of the model consists of three parts: encoder, age condition, and decoder. The encoder is used for feature extraction of the input face image. We are using three groups of convolution layer and nine residual blocks. Each convolution layer consists of a ReLU non-linearity layer, spatial batch normalization layer stride, and a convolution layer. After this, we get a 56 feature map of the image. Since there is a huge range of images of a large number of age groups in our dataset we are dividing the image into y number of groups which is represented as a one-hot age vector. Each block represents a specific age group. The task of the decoder is to convert the face to generate an image of the required age group based on the features.

3.2 Proposed Methodology

3.2.1 Age-cGAN:

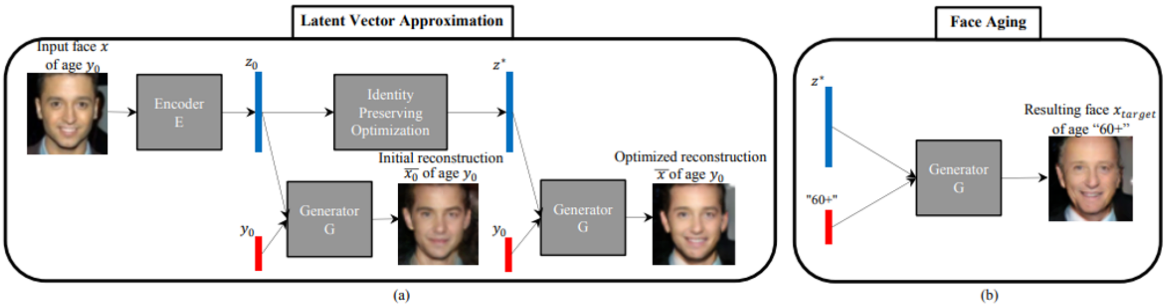


Figure 2: Face Aging Method. (a) approximation of the latent vector to reconstruct the input image; (b) switching the age condition at the input of the generator G to perform face aging. [5]

The Face Aging-cGAN has four networks:

Encoder: It is used to learn the inverse mapping of input image and age condition with the latent vector Z.

- Encoder network is a Convolutional Neural Network that generates a latent vector of 100 dimensions of the input face image of dimension (64,64,3).

- There are 2 dense layers and 4 convolution blocks.
- All the convolution blocks except the first layer has a convolution layer followed by a batch normalization layer and then an activation function.

FaceNet: It is the face recognition model that learns the difference between the original image x (input image) and the generated image x'.

- For FaceNet Inception ResNet, ResNet-50, or pre-trained Inception can be used. FaceNet is used to recognize a person's identity in the input image. The extracted embeddings for the original input image and reconstructed image can be found by calculated Euclidean distance.

Generator Network: It takes a face image and condition vector and tries to generate realistic images.

- It is a CNN having upsampling, convolutional, and dense layers which take a condition vector and a latent vector to generate a realistic image of a dimension of (64, 64, 3). The condition vector is the additional information that is provided to the network. For the Age-cGAN, this will be the age.

Discriminator Network: It is a CNN that is used to discriminate between the real image and the fake image.

3.2.2 FaceNet:

We are using FaceNet in our work for face recognition. It uses Convolution layers to learn embeddings from the face directly which are used for face verification and face recognition. These embeddings are of 128 dimensions and insert them into feature space such that the squared distance between faces of the same person is less whereas the squared distance between images to two different people is large.

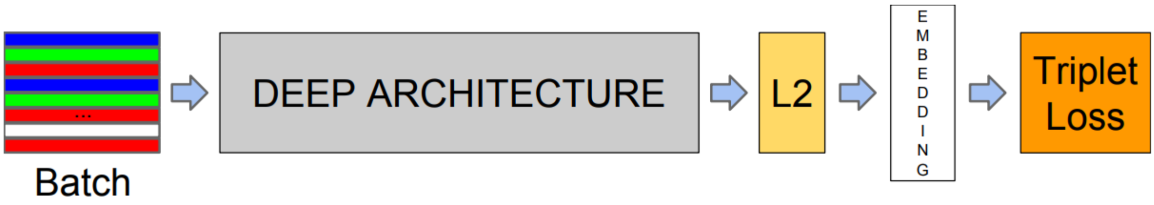


Figure 3: Model architecture of FaceNet.

3.2.3 FG-NET Dataset:

For our age progression module, we are going to use the FG-NET aging dataset which is a large-scale face dataset containing images of people and children with a long age span ranging from 0 to 40 years of age. The Face and Gesture Network(FG-NET) database was released in the year 2004 which was used by many researchers with the aim of understanding the change in facial appearance which is caused due to aging. The dataset contains more than twenty thousand images of a face with annotation of ethnicity, gender, and age. The dataset contains images with huge variations in facial expression, illumination, resolution, occlusion, pose, etc. The dataset also provides the corresponding landmarks. This dataset can be used for a variety of tasks including age estimation, age progression, landmark localization, progression/ regression, face detection, etc.

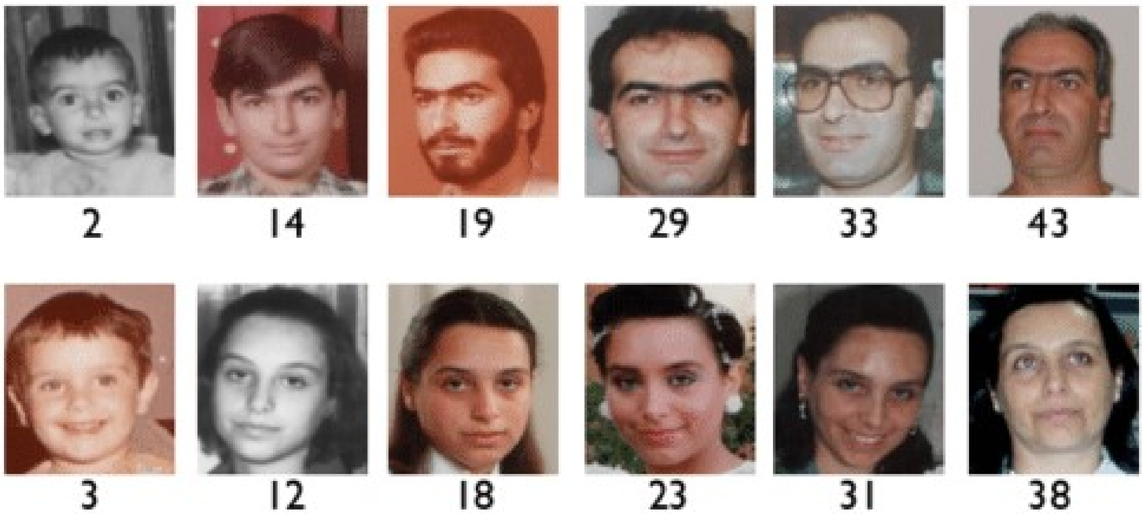


Figure 4: Sample images of different individuals from FG-NET face dataset

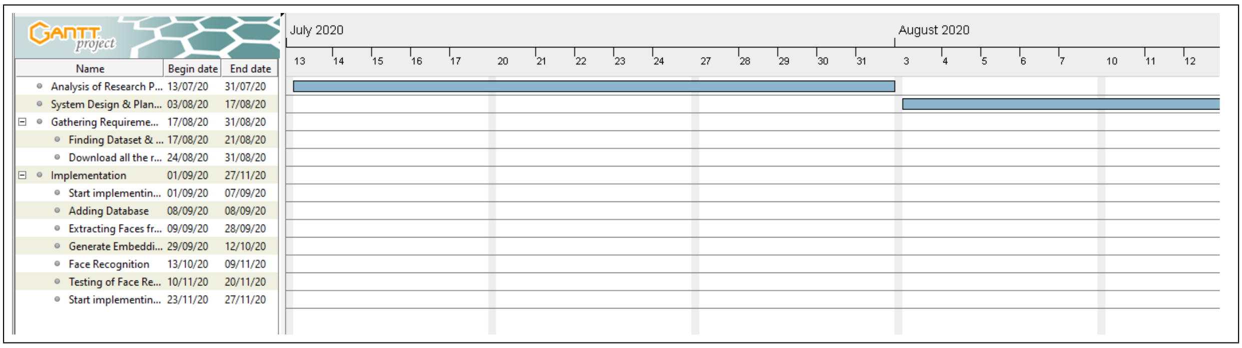
3.3 Details of Hardware/Software Requirement

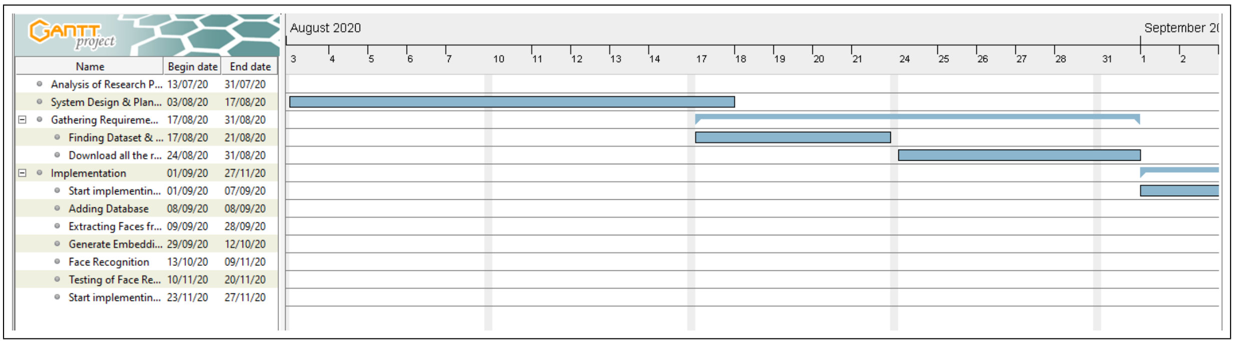
- **python 3.8.5**
- **Anaconda v4.8.3**: Conda is an open-source distribution of R and Python programming language for data science and scientific computation which is used to simplify deployment and package management.
- **Jupyter Notebook**: The Jupyter Notebook is an open-source web application that is used for you to share and create documents that contain narrative text, visualization, equations, live code. Uses include machine learning, data visualization, statistical modeling, numerical simulation data cleaning and transformation, and much more.
- **NumPy 1.19.2**: NumPy is a library that is used for working with large, multidimensional matrices arrays.
- **pandas 1.1.3**: pandas is a fast and powerful open-source data manipulation and analysis tool.
- **scikit-image 0.17.2**
- **matplotlib 3.3.2**
- **Keras** is written in Python and it is one of the leading high-level neural network APIs which supports multiple back-end neural network computation engines.
- **Tensorflow v2.3.1:** TensorFlow is a flexible and comprehensive end-to-end open-source platform for machine learning.

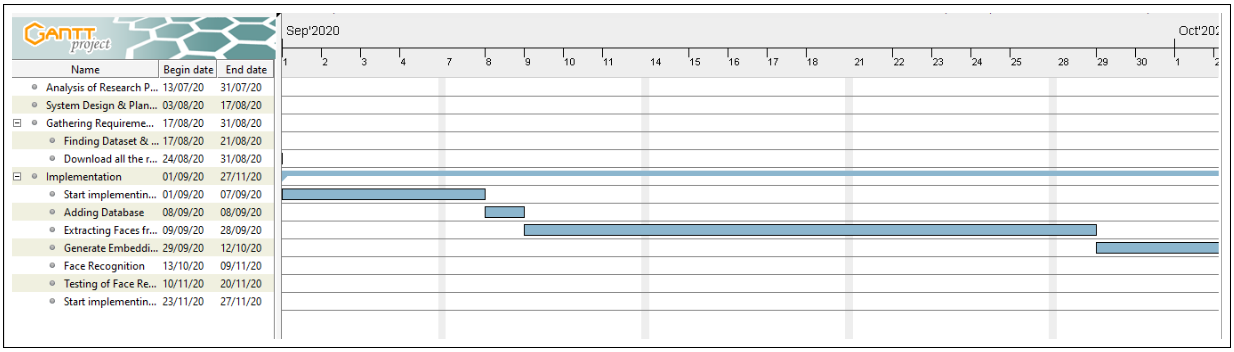
Chapter 4

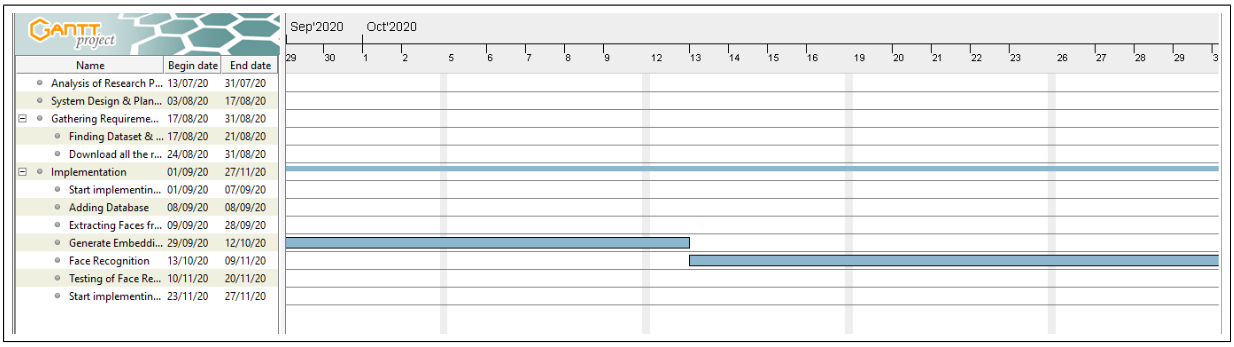
Planning And Formulation

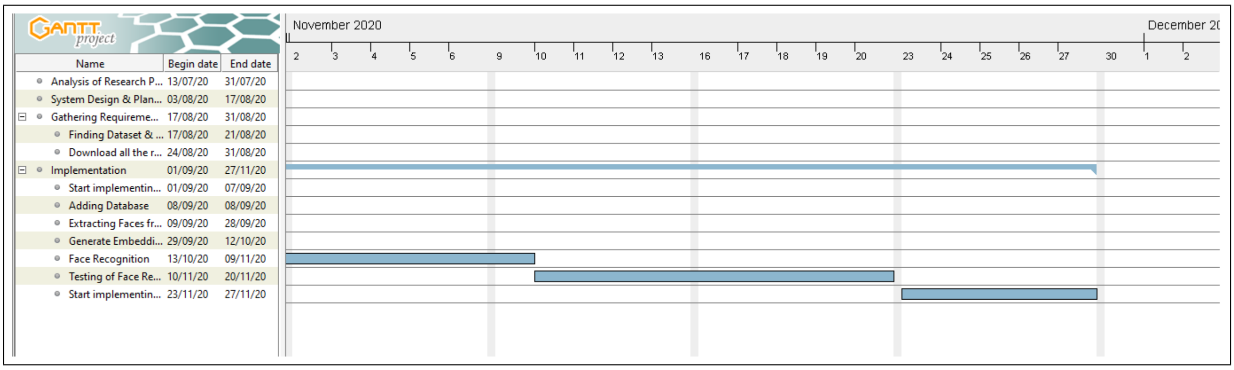
4.1 Schedule for Project






Sep'2020
Oct'2020
Name
Begin date
End date
Analysis of Research P... 13/07/20 31/07/20
System Design & Plan... 03/08/20 17/08/20
Gathering Requireme... 17/08/20 31/08/20
Finding Dataset & ... 17/08/20 21/08/20
Download all the r... 24/08/20 31/08/20
Implementation 01/09/20 27/11/20
Start implementin... 01/09/20 07/09/20
Adding Database 08/09/20 08/09/20
Extracting Faces fr... 09/09/20 28/09/20
Generate Embeddi... 29/09/20 12/10/20
Face Recognition 13/10/20 09/11/20
Testing of Face Re... 10/11/20 20/11/20
Start implementin... 23/11/20 27/11/20


Sep'2020
Oct'2020
Name
Begin date
End date
Analysis of Research P... 13/07/20 31/07/20
System Design & Plan... 03/08/20 17/08/20
Gathering Requireme... 17/08/20 31/08/20
Finding Dataset & ... 17/08/20 21/08/20
Download all the r... 24/08/20 31/08/20
Implementation 01/09/20 27/11/20
Start implementin... 01/09/20 07/09/20
Adding Database 08/09/20 08/09/20
Extracting Faces fr... 09/09/20 28/09/20
Generate Embeddi... 29/09/20 12/10/20
Face Recognition 13/10/20 09/11/20
Testing of Face Re... 10/11/20 20/11/20
Start implementin... 23/11/20 27/11/20


November 2020
Name
Begin date
End date
Analysis of Research P... 13/07/20 31/07/20
System Design & Plan... 03/08/20 17/08/20
Gathering Requireme... 17/08/20 31/08/20
Finding Dataset & ... 17/08/20 21/08/20
Download all the r... 24/08/20 31/08/20
Implementation 01/09/20 27/11/20
Start implementin... 01/09/20 07/09/20
Adding Database 08/09/20 08/09/20
Extracting Faces fr... 09/09/20 28/09/20
Generate Embeddi... 29/09/20 12/10/20
Face Recognition 13/10/20 09/11/20
Testing of Face Re... 10/11/20 20/11/20
Start implementin... 23/11/20 27/11/20

4.2 Detail Plan of Execution

- **Existing System Study:** At the start of July, We started gathering the research papers that are related to the Lost child identification topic. After going through many research papers, we have done the analysis of papers and Literature survey through which we come to know that there are some limitations present in the existing system.

- **System Design and Planning:** At the start of August, We started designing our system after analyzing the existing system. To overcome the limitations of the existing system we decided to work on the Aging invariant module in face recognition. After much research, we decided to use a Conditional- Generative Adversarial Network and we decided the further planning of our system.

- **Gathering System Requirements:** From the middle of August week, we started gathering the Child dataset that is required for the system. After testing some child datasets, we start creating our dataset from scratch for more accuracy. For our system as we are going to use Keras and Tensorflow, we started downloading all the important libraries that are required for implementation.

- **Implementation:** At the start of September, We started the implementation process of our system. We started implementing the Face recognition module. First, we loaded the dataset that we created from scratch into the Jupyter notebook that we created from scratch. After that, we started implementing the module which extracts the faces of children from images and labels them, and stored them in a different folder. After completing the Extracting face module, we started to generate embeddings of images and after that, we started work on face recognition. After completing the face recognition module we started testing. At the start of testing, we got many errors and false predictions but after training the dataset we got accurate predictions for each image. At end of November, we started implementing the Aging invariant module.

Chapter 5

Design of System

5.1 Design Diagram with Explanation

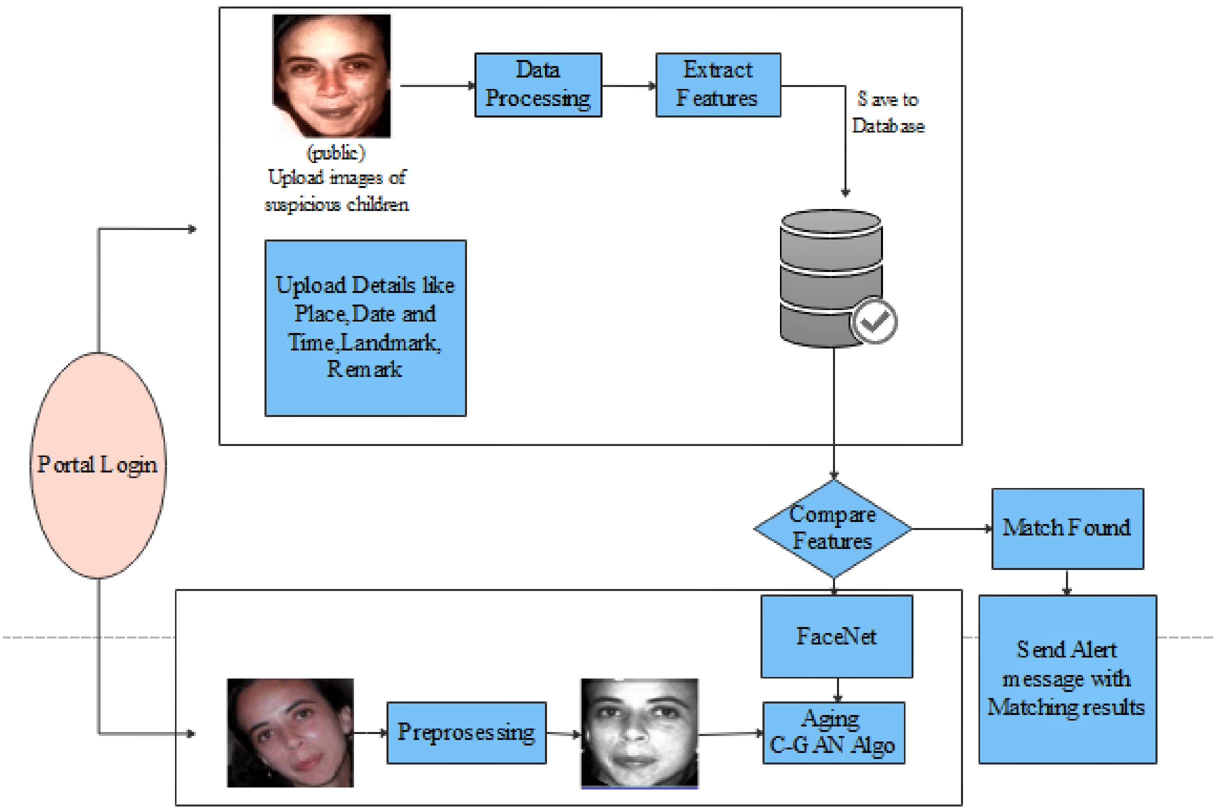


Figure 5: System design for lost child identification using FaceNet and C-GAN

The proposed system will consist of a portal where the public can store the details with a photograph of the missing child. The public here refers to the parent of the missing child or the police officer. By using the photograph the public can search for matching children from the already stored data. The system will prompt the most matching cases after applying a face recognition algorithm to the uploaded image. Once the matching is done the concerned person can get the details of the child. The proposed model is divided into various phases. The first step is the image preprocessing phase. The input to this is the raw image that is uploaded by the public. And the output of this phase is an aligned frontal face image. The main purpose of this step is to align all the face images based on eye coordinates such that all the images are aligned with the same standard size and to register for the database.

The next stage is to extract the required face features. This stage uses a Multi-task Cascaded Convolutional Networks (MTCNN) model which is a framework used for face alignment and face detection which consists of three stages of convolutional networks that can detect landmark location such as eyes, nose, and mouth and can recognize face very efficiently. The preprocessing step includes removing noise from the image, straighten the image, and detecting and cropping the face from the whole image. The extracted face features will be stored in the database for further processing. Now, whenever the public finds and missing or suspicious child they can click and upload the photograph of the child in the portal. The public also needs to upload details like the name of the child, age, place where the child was found, date and time when the child was found. The public can also add remarks like what clothes the child was wearing when he was spotted or any information the child might have told the person which will help to find the child. This uploaded image will go through the age C-GAN algorithm after the preprocessing and feature extraction step. The GAN algorithm is used to generate a new image with the same statistics as the training images. As compared to the previous works using C-GAN for changing the face features and attributes, we are focusing more on preserving the identity of the original child in his/her age-progressed image. So basically this model will be used to synthesis child face to required age categories. This face synthesized and age-progressed image goes through a FaceNet for feature extraction and face recognition. The main distinction between FaceNet and other face recognition algorithms is that faceNet does not use any bottleneck layer for verification and recognition tasks but rather learns the mapping directly from images and creates embedding. The FaceNet is used to automatically compare the age-progressed photo of the missing child with the already present images in the database and if the photo matches with any of the photos present in the database then an alert message will be sent to the parent of the missing child along with various required details like name and location where the child was found.

Chapter 6

Expected Results

6.1 Implementation Details

We started our implementation by creating a Face Recognition module using FaceNet. Face recognition is the process of identifying and verifying the person from images of their faces. First, we created the dataset from scratch containing many photos of children. For implementation, we downloaded all required Libraries that are scikit-image 0.17.2 , matplotlib 3.3.2, NumPy 1.19.2 , pandas 1.1.3, Tensorflow v2.3.1, Keras, and also we downloaded the FaceNet model which is required for Keras. To detect faces for Face recognition we use the Multi-Task Cascaded Convolutional Neural Network or MTCNN.

Loading the Dataset : We created a dataset containing many images of children. To train the dataset first we have to load it in a library. After giving the path to a directory of a dataset, each photograph present in the dataset loaded along with the shape of a NumPy array containing face pixel data.

Detecting and Extracting Faces : The first step is to detect faces in each photograph present in the dataset and reduce the dataset by placing only the faces of children. The extractface( ) function is to extract the faces from each photograph. Our dataset is consist of 11 children's photos from extract face function We can detect the face in each photograph, and create a plot with each face, Each face was correctly detected, and that we have a range of lighting, skin tones, and orientations in the detected faces. For extraction of faces, we prepare a dataset with the name as the output label for each detected face. The generatefacesfromimages( ) function loaded all of the faces into a list for a given directory. After calling the generatefacesfromimages( ) function, Each face has one label, the name of the children, which we take from the directory name loaded. The generatefacesfromimages function detects faces for each subdirectory of children assigning labels to each detected face and stored it in a different directory. We can then call the function for the 'minidataset' and 'children duplicate' folders to load all of the data, then save the results in a single compressed NumPy array file via the savezCompressed() function.

After running the extractFace() function, First, all of the photos in the 'minidataset' dataset are loaded, then faces are extracted, resulting in 14 samples with square face input and a class label string as output. Then the 'children duplicate' dataset is loaded, providing 13 samples that can be used as a test dataset. Both datasets are then saved to a compressed NumPy array file called faceDatasetNumpy.npz ' that is about three megabytes and is stored in the current working directory.

Generating Embedding from faces : The next step is to create Embeddings from faces. A face embedding is a vector that represents the features extracted from the face. This can then be compared with the vectors generated for other faces. For example, another vector that is close by some measure may be the same person, whereas another vector that is far by some measure may be a different person. The classifier model that we want to develop will take a face embedding as input and predict the identity of the face. The FaceNet model generates this embedding for a given image of a face. We used the FaceNet model to pre-process a face to create a face embedding that can be stored and used as input to our classifier model. We can, therefore, pre-compute the face embeddings for all faces in the 'minidataset' and children duplicate dataset. After running the getEmbedding() function face dataset was loaded correctly and so was the model. The training dataset was then transformed into 23 face embeddings, each comprised of a 128 element vector. The 23 examples in the 'minidatset' dataset were also suitably converted to face embeddings. The generated dataset were then saved to a compressed NumPy array with the name 'faceEmbeddings.npz ' in the current working directory.

Classification, Normalization, and Prediction : For Face recognition, first, we have to classify the face embeddings. We have to load the face embeddings to classify. The data require some minor preparation concerning modelling for that we have to normalize the face embedding vectors because the vectors are often compared to each other using a distance metric. Vector normalization means scaling the values until the length or magnitude of the vectors is one of unit length. For Vector normalization, we used the Normalizer class in scikit-learn with the face embeddings. After that string target variables for each child's name need to be converted to integers for that Labalization is required. For Labelization we used LabelEncoder class in scikit-learn. The next step is prediction, We have to create the dataset which contains the photographs that we want to predict. After creating the predicted dataset we have generated their embeddings for faces by using the 'faceEmbedding.npz' file. After that, we have to load any image that we want to predict and after extracting the face from that image it will generate its closed prediction. Every child we want to recognize having their directory with their images in it. The images contain the face of only one person. If the image contains multiple faces, only the one detected with the highest probability will be considered and accordingly, it will generate the given image's closed prediction.

6.2 Project Outcomes

First load all required libraries.

```
In [1]: from os import listdir
        from os.path import isdir
        from PIL import Image
        from matplotlib import pyplot
        from numpy import savez_compressed
        from numpy import asarray
        from mtcnn.mtcnn import MTCNN
        import time
        import os
        import numpy as np
        os.environ["KMP_DUPLICATE_LIB_OK"]="TRUE"
        from pathlib import Path

        import neha
```

Load the dataset:

```
In [2]: persons_dataset_path ="C:/children/minidataset"
        images, labels=neha.load_rgb_data(persons_dataset_path,200,shuffle=True)

Loading images...
['Amisha', 'arnav', 'Chakuli', 'Gaurangi', 'Megh', 'Piyush', 'Pratiksha', 'Rutuja', 'Smiti', 'Soham', 'Sushmita', 'Swarali', 'Y
og']
Loading Amisha
we will load [ 2 ] files from [ Amisha ] class ...
Loading arnav
we will load [ 2 ] files from [ arnav ] class ...
Loading Chakuli
we will load [ 2 ] files from [ Chakuli ] class ...
Loading Gaurangi
we will load [ 1 ] files from [ Gaurangi ] class ...
Loading Megh
we will load [ 2 ] files from [ Megh ] class ...
Loading Piyush
we will load [ 2 ] files from [ Piyush ] class ...
Loading Pratiksha
we will load [ 2 ] files from [ Pratiksha ] class ...
Loading Rutuja
we will load [ 2 ] files from [ Rutuja ] class ...
Loading Smiti
we will load [ 1 ] files from [ Smiti ] class ...
Loading Soham
we will load [ 2 ] files from [ Soham ] class ...
Loading Sushmita
we will load [ 2 ] files from [ Sushmita ] class ...
Loading Swarali
```

Detect the faces from the images present in the dataset and extract faces from images and label each extracting face are done in the extractFace() function.

```
In [4]: def extract_face(path_to_filename, detector, required_size=(160,160), save_faces=True):
            image=Image.open(path_to_filename)
            image=image.convert('RGB')
            pixels=asarray(image)
            results=detector.detect_faces(pixels)
            x1,y1,width,height=results[0]['box']
            x1,y1 = abs(x1),abs(y1)
            x2,y2 = x1 + width, y1 + height
            face = pixels[y1:y2, x1:x2]
            image = Image.fromarray(face)
            image = image.resize(required_size)
            if(save_faces):
                path = os.path.split(os.path.abspath(path_to_filename))[0]
                file_name = os.path.split(os.path.abspath(path_to_filename))[1]
                person_name = os.path.basename(os.path.normpath(Path(path)))
                project_folder = Path(path).parent.parent
                print(person_name)
                target_folder = os.path.join(project_folder, 'children_duplicate' ,person_name)
                if not os.path.exists(target_folder):
                    os.makedirs(target_folder)
                target_face_file_path = os.path.join(target_folder, file_name)
                print(target_face_file_path)
                image.save(target_face_file_path)
            face_array = asarray(image)
            return face_array
```

```
In [5]: def extract_faces(directory):
            print('load faces')
            faces = list()

            detector=MTCNN()
            print('Extracting faces from ', directory, '...')
            for filename in listdir(directory):
                path = directory + filename
                try:
                    face = extract_face(path, detector, save_faces=True)
                except Exception as e:
                    continue
                faces.append(face)
            return faces
```

```
In [6]: def generate_faces_from_images(directory):
            print('Load dataset ...')
            X,y = list(),list()
            num = 1
            for subdir in listdir(directory):
                path = directory + '/' + subdir + '/'
                if not isdir(path):
                    continue
                faces = extract_faces(path)
                labels = [subdir for _ in range(len(faces))]
                print('> %d) loaded %d examples for class : %s' % (num , len(faces), subdir))
                num =num + 1
                X.extend(faces)
```

All the photos in the 'minidataset' dataset are loaded, then faces are extracted, resulting in 14 samples with square face input and a class label string as output, and all these labelled extract faces are stored in a different directory called 'Children duplicate'. Then the 'children duplicate' dataset is loaded, providing 13 samples that can be used as a test dataset. Both datasets are then saved to a compressed NumPy array file called 'faceDatasetNumpy.npz ' that is about three megabytes and is stored in the current working directory.

```
In [7]: faces, labels = generate_faces_from_images(persons_dataset_path)
        print(faces.shape, labels.shape)

        Load dataset ...
        load faces
        Extracting faces from  C:/children/minidataset/Amisha/ ...
        Amisha
        C:\children\children_duplicate\Amisha\amisha.jpeg
        Amisha
        C:\children\children_duplicate\Amisha\ammu.jpeg
        > 1) loaded 2 examples for class : Amisha
        load faces
        Extracting faces from  C:/children/minidataset/arnav/ ...
        arnav
        C:\children\children_duplicate\arnav\arnav.jpeg
        arnav
        C:\children\children_duplicate\arnav\arnav1.jpeg
        > 2) loaded 2 examples for class : arnav
        load faces
        Extracting faces from  C:/children/minidataset/Chakuli/ ...
        Chakuli
        C:\children\children_duplicate\Chakuli\Chakuli-001.jpeg
```

```
In [8]: savez_compressed("face_dataset_numpy.npz" , faces, labels)
```

After extracting faces from children photos present into 'minidataset'. Images that are stored in ‘children duplicate’ directory :

```
In [10]: neha.plot_sample_from_dataset(images,labels,rows=1,colums=8,width=20,height=20)
```


```
In [11]: faces_dataset_numpy= "face_dataset_numpy.npz"
         from numpy import load
         data = load(faces_dataset_numpy)
         faces, labels2 = data['arr_0'], data['arr_1']
```

```
In [12]: import matplotlib.pyplot as plt

         neha.display_image(images, labels, index=7)

         Label = Rutuja
         image shape:  (200, 200, 3)
```

Generating embedding from faces that are stored in a dataset is done using the getEmbedding() function. After running the getEmbedding() function face dataset was loaded correctly and so was the model. The training dataset was then transformed into 23 face embeddings, each comprised of a 128 element vector. The 23 examples in the minidatset dataset were also suitably converted to face embeddings. The resulting datasets were then saved to a compressed NumPy array that is about 50 kilobytes with the name ’faceEmbeddings.npz ’ in the current working directory.

```
In [13]: #generate embedding from faces
from numpy import load
from numpy import expand_dims
from numpy import asarray
from numpy import savez_compressed
import tensorflow as tf
```

```
In [14]: def get_embedding(model, face_pixels):
    face_pixels = face_pixels.astype('float32')
    mean, std = face_pixels.mean(), face_pixels.std()
    face_pixels = (face_pixels - mean) / std
    samples = expand_dims(face_pixels, axis=0)
    yhat = model.predict(samples)     return yhat[0]
```

```
In [15]: faces_dataset_path= "face_dataset_numpy.npz"
data = load(faces_dataset_path)
faces, labels = data['arr_0'], data['arr_1']
print('Loaded: ', faces.shape, labels.shape)

Loaded:  (23, 160, 160, 3) (23,)
```

```
In [16]: model=tf.keras.models.load_model('facenet_keras.h5')
print('Loaded Model')

WARNING:tensorflow:No training configuration found in the save file, so the model was *not* compiled. Compile it manually.
Loaded Model
```

```
In [17]: face_embeddings = list()
for face_pixels in faces:
    embedding=get_embedding(model, face_pixels)
    face_embeddings.append(embedding)
face_embeddings = asarray(face_embeddings)
print(face_embeddings.shape)

(23, 128)
```

```
In [18]: savez_compressed('face_embeddings.npz', face_embeddings, labels)
```

For face recognition we import all the required libraries and FaceNet model :

```
In [19]: #FACE RECOGNITION
import tensorflow as tf
from numpy import asarray
from numpy import load
import time
from detect_face_from_one_image import extract_face
from image_embeddings import get_embedding
from mtcnn.mtcnn import MTCNN
import re
from os import listdir
import os
import neha
import csv
```

```
In [20]: tick = time.time()
model = tf.keras.models.load_model('facenet_keras.h5')
print('Time to load facenet model and dataset : ', time.time()-tick, "sec")
print('********* Loaded Model **********')

WARNING:tensorflow:No training configuration found in the save file, so the model was *not* compiled. Compile it manually.
Time to load facenet model and dataset :  4.773072242736816 sec
********* Loaded Model **********
```

Classification, Normalization, and Prediction :

```python
In [21]: import numpy
from numpy import expand_dims
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import Normalizer
from sklearn.svm import SVC
from matplotlib import pyplot
from sklearn.neighbors import KNeighborsClassifier
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import accuracy_score
from sklearn.metrics import roc_curve
from numpy import load
from sklearn.metrics import classification_report

def classification(faces_embedding, labels):
    faces_embedding = normalization2(faces_embedding)
    labels = lablization(labels)

    classifier = KNeighborsClassifier(n_neighbors=8, p=1, weights="distance", metric="euclidean")
    classifier.fit(faces_embedding, labels)

    return classifier


def normalization(faces_embedding, face_to_predict_embedding):
    in_encoder = Normalizer(norm='l2')
    faces_embedding = in_encoder.transform(faces_embedding)
    face_to_predict_embedding = in_encoder.transform(face_to_predict_embedding)

    return faces_embedding, face_to_predict_embedding
```

```python
def lablization(labels) :

    out_encoder = LabelEncoder()

    out_encoder.fit(labels)

    labels = out_encoder.transform(labels)

    return labels

def predict2(face_to_predict_embedding, faces_embedding, labels, classifier) :
    out_encoder = LabelEncoder()
    out_encoder.fit(labels)

    labels = out_encoder.transform(labels)

    faces_embedding, face_to_predict_embedding = normalization(faces_embedding, face_to_predict_embedding)

    face_emb = face_to_predict_embedding[0]

    #prediction for the faces
    samples = expand_dims(face_emb, axis=0)
    yhat_class = classifier.predict(samples)
    print(yhat_class)
    yhat_prob = classifier.predict_proba(samples)
    print(yhat_class)

    class_index = yhat_class[0]
    class_probability = yhat_prob[0,class_index] * 100

    predict_names = out_encoder.inverse_transform(yhat_class)[0]

    return predict_names, class_probability
```

Insert the input image that we want to predict and extract face and generate embeddings of that input image:

In [28]:

```
#Load dataset
faces_embeddings_path = 'face_embeddings.npz'
data = load(faces_embeddings_path)
faces_embedding, labels = data['arr_0'], data['arr_1']

detector = MTCNN()
tick = time.time()
classifier = classification(faces_embedding, labels)
print("classifier time: ", time.time()-tick)
print("************* Loaded Embeddings **************")
```

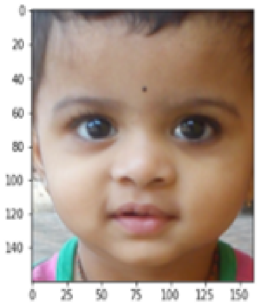
```
classifier time:  0.034960031509399414
************* Loaded Embeddings **************
```

In [68]:

```
#extract faces from image
tick = time.time()
image_to_predict = "C:/children/FACEREC/recognization/Gauu.jpg"
image = extract_face(detector, image_to_predict)

neha.display_one_image(image, image_to_predict)
print('Time to extract image : ',time.time()-tick)
```

```
Label = C:/children/FACEREC/recognization/Gauu.jpg
image shape:  (160, 160, 3)
Time to extract image :  11.723385572433472
```



Generates its closed Prediction :

In [27]:

```
#predict2
tick = time.time()

predict_name, class_probability = predict2(face_to_predict_embedding, faces_embedding, labels, classifier)

unknown_name = "unknown"

if(class_probability >= 27) :
    print('Predicted : %s (%.3f)' % (predict_name,class_probability))
    unknown_name= ' '

else :
    unknown_name = "Unknown"
    print('Predicted : %s (%.3f)'  % (unknown_name, class_probability))
    print('Closed Prediction : ',predict_name)

face_expected_name = re.findall("facenet/(\w+).jpg", image_to_predict)
print(face_expected_name)

print("Total time to Predict: ",time.time()-tick)
```

```
[2]
[2]
Predicted : Gaurangi (28.806)
[]
Total time to Predict:  0.17674040794372559
```

Chapter 7

Conclusion and Future Work

7.0.1 Conclusion

As the traumatic conditions appeared in the identification of missing child the process gets much and much harder. And this results in the increasing number of missing child cases among them many have occured because identifying a missing child with a photograph is harder as the age gap is also one of the important aspects in the identification and this aspect creates many difficulties in the process of identification and considering all these aspects and difficulties our system will solve all these problems with the deep learning and age gap feature. By using the FaceNet algorithm which will give a perfect result in identifying any child with the stored photograph also it solves the problem of an age gap with the CGan algorithm which will help to identify any child even after the 5 or 10 years when he lost. It will be very beneficial to solve the number of cases of the untraced missing child and will help to reduce the increasing number of cases of the missing children.

7.0.2 Future Work

Generative approaches are found effective in facial form modeling, whereas, age invariant signatures are effectively extracted by discriminative strategies. Hence, a joint approach of world and native options might lead to higher performance because the human cognitive process is an integration of each approach. Recently, it's been unconcealed that 3D facial image options square measure even additional reliability than blood profile for distinguishing age. Also, aging introduces facial deformations in 3D space, so developing generative still as discriminative strategies supported 3D face models need to be explored additional. Facial aging is stricken by numerous intrinsic still as alien factors, however, their qualitative analysis is missing. additionally, analysis of the impact of gender and race is additionally a prominent future direction within the development of a strong face recognition system. Therefore, coming up with strong age invariant options and corresponding generalizable matching framework effective across aging and non-aging situations each may be a viable future direction.

References

[1] S. Chandran, Pournami Balakrishnan, Byju Rajasekharan, Deepak N Nishakumari, K Devanand, P M Sasi, P. (2018). “Missing Child Identification System Using Deep Learning and Multiclass SVM”.

[2] Xin Jin, Shiming Ge, Chenggen Song, Xiaodong Li, Jicheng Lei, Chuanqiang Wu, and Haoyang Yu “Double-Blinded Finder: A Two-SidePrivacy-Preserving Approach for Finding Missing Children”(2020).

[3] Shun Lei Myat Oo, Aung Nway Oo University of Information Technology, Yangon, Myanmar “ Child Face Recognition with Deep Learning”(2019).

[4] Chang Shu School of Communication and Information Engineering,University of Electronic Science and Technology of China Chengdu,China “Optimizing deep neural network structure for face recognition” (2017).

[5] Grigory Antipov, Moez Baccouche, Jean-Luc Dugelay ”Face Aging with Conditional Generative Adversarial Networks” (2017) arXiv:1702.01983v2

Appendices

Appendix A

Weekly Progress Report Project A

Appendix B

Plagiarism Report

F_34-Identifying lost children using deep learning

ORIGINALITY REPORT

32%

SIMILARITY INDEX

PRIMARY SOURCES

1	machinelearningmastery.com Internet	592 words — 8%
2	Pournami S. Chandran, N B Byju, R U Deepak, K N Nishakumari, P Devanand, P M Sasi. "Missing Child Identification System Using Deep Learning and Multiclass SVM", 2018 IEEE Recent Advances in Intelligent Computational Systems (RAICS), 2018 Crossref	281 words — 4%
3	"3rd EAI International Conference on Robotic Sensor Networks", Springer Science and Business Media LLC, 2021 Crossref	153 words — 2%
4	www.itm-conferences.org Internet	132 words — 2%
5	Chang Shu, Hongsheng Liu, Fanruo Meng. "Optimizing deep neural network structure for face recognition", 2017 IEEE International Symposium on Circuits and Systems (ISCAS), 2017 Crossref	127 words — 2%
6	Shun Lei Myat Oo, Aung Nway Oo. "Child Face Recognition with Deep Learning", 2019 International Conference on Advanced Information Technologies (ICAIT), 2019 Crossref	123 words — 2%
7	Amal A. Moustafa, Ahmed Elnakib, Nihal F. F. Areed. "Age-invariant face recognition based on deep features analysis", Signal, Image and Video Processing, 2020	115 words — 2%

Acknowledgement

We take this opportunity to express my profound gratitude and deep regards to our guide **Mrs. Smita Bharne** for his/her exemplary guidance, monitoring and constant encouragement throughout the completion of this report. We truly grateful to his/her efforts to improve our understanding towards various concepts and technical skills required in our project. The blessing, help and guidance given by him/her time to time shall carry us a long way in the journey of life on which we are about to embark.

We take this privilege to express our sincere thanks to **Dr. Mukesh D. Patil**, Principal, RAIT for providing the much necessary facilities. We are also thankful to **Dr. Leena Ragha**, Head of Department of Computer Engineering, Project Co-ordinator **Mrs. Smita Bharne** and Project Co-coordinator **Mrs. Bhavana Alte**, Department of Computer Engineering, RAIT, Nerul Navi Mumbai for their generous support.

Last but not the least we would also like to thank all those who have directly or indirectly helped us in completion of this thesis.

Ms. Amisha Mangesh Pupala
Ms. Samruddhi Manohar Mokal
Ms Neha Ananta Pandit