Project Report

On

” Face Recognition using Graph Auto Encoder”

In fulfillment of requirements for the award of the degree in
Master of Computer Applications 2023

Master of Computer Applications

Submitted by

Krishanu Bhattacharjee	**Ram Babu Ray**	**Rashmi Keot**
Roll No: 20MCA006	Roll No: 20MCA013	Roll No: 20MCA014

Under the guidance of
Dr. Swarup Roy
Professor
of Computer Applications Sikkim
University, Gangtok-737102, India

Department of Computer Applications
Sikkim University
2023

Dedicated to
Father & Mother

Acknowledgements

It was our privilege and honor to work under the supervision of Dr. Swarup Roy, Professor of Computer Applications, Sikkim University, Gangtok, India. We would like to express our sincere gratitude to our supervisor Dr. Swarup Roy for his excellent guidance, care, patience, and for providing us with an excellent atmosphere for doing research.

We would like to thank all faculty members at the Department of Computer Applications, Sikkim University, Gangtok, India, for their valuable suggestions during our research work. I would like to thank the staffs of the Department of Department of Computer Applications, Sikkim University, Gangtok, India, who were always willing to help and give their best support in our project work journey.

We deeply acknowledge the love, cooperation, and the moral support extended by our parents, friends, relatives, and colleagues right from the beginning of project work.

Finally, we must thank the Almighty for blessing us in this journey.

Krishanu Bhattacharjee	**Ram Babu Ray**	**Rashmi Keot**
Roll No: 20MCA006	Roll No: 20MCA013	Roll No: 20MCA014

Declaration

I **Krishanu Bhattacharjee [Roll No. 20MCA006], Ram Babu Ray, [Roll No. 20MCA013], Rashmi Keot [Roll No. 20MCA014]**, a registered candidate for Master of Computer Applications under Department of Computer Applications (MCA) of Sikkim University, Gangtok, India, declare that this is my own original work and does not contain material for which the copyright belongs to a third party and that it has not been presented and will not be presented to any other University/ Institute for a similar or any other Degree award.

I further confirm that for all third-party copyright material in my project report (including any electronic attachments), I have "blanked out" third party material from the copies of the thesis fully referenced the deleted materials and where possible, provided links (URL) to electronic sources of the material.

I hereby transfer exclusive copyright for this project report to Sikkim University. The following rights are resewed by the author:

a) The right to use, free of charge, all or part of this article in future work of their own, such as books and lectures, giving reference to the original place of publication and copyright holding.

b) The right to reproduce the article or thesis for their own purpose provided the copies are not offered for sale.

Krishanu

Krishanu Bhattacharjee
Roll No: 20MCA006

Rambabu Ray

Ram Babu Ray
Roll No: 20MCA013

R Keot

Rashmi Keot
Roll No: 20MCA014

Certificate from the Supervisor

This is to certify that Mr. **Krishanu Bhattacharjee, [Roll No. 20MCA006], Ram Babu Ray, [Roll No. 20MCA013] and Rashmi Keot, [Roll No. 20MCA014]** are registered candidate for Master of Computer Applications Program under the department of **Computer Applications** of Sikkim University, Gangtok, India.

The undersigned certifies that they have completed all other requirements for submission of the major project and hereby recommends for the acceptance of their project entitled' **Face Recognition using Graph Auto Encoder** in the partial fulfillment of the requirements for the award of MCA Degree by Sikkim University, Gangtok, India.

Dr. Swarup Roy

Supervisor, Professor

Department of Computer Applications
Sikkim University Gangtok

Date:

Certificate from the Head of Department

This is to certify that Mr. **Krishanu Bhattacharjee, (Roll No. 20MCA006), Ram Babu Ray, (Roll No. 20MCA013) and Rashmi Keot, (Roll No. 20MCA014)** are registered candidate for Master of Computer Applications Program under the department of **Computer Applications** of Sikkim University, Gangtok, India.

The undersigned certifies that they have completed all other requirements for submission of the major project and hereby recommends for the acceptance of their project entitled' **Face Recognition using Graph Auto Encoder** in the partial fulfillment of the requirements for the award of MCA Degree by Sikkim University, Gangtok, India.

Dr. Mohan Pratap Pradhan

Associate Professor and Head

Department of Computer Applications
Sikkim University Gangtok

Date:

Abstract

Face recognition is a rapidly growing field of computer vision that aims to automatically identify individuals by analysing their facial features. It has numerous applications in security, surveillance, and access control systems, as well as in social media and online services that require identity verification.

The goal of a face recognition system is to match a person's face to a pre-existing database of faces, or to identify unknown faces in real-time. This is typically achieved by extracting facial features such as the distance between the eyes, the shape of the jawline, and the contours of the nose and mouth, and comparing them to a database of previously stored facial features.

With recent advancements in deep learning and neural network models, face recognition systems have achieved unprecedented levels of accuracy and robustness. However, the use of such systems raises concerns about privacy, bias, and ethical issues, which need to be addressed and mitigated.

In this project, we will explore the fundamentals of face recognition, and develop a deep learning-based model to recognize faces in images and videos. We will use state-of-the-art techniques such as convolutional neural networks, transfer learning, and data augmentation to train our model on a large dataset of faces. Finally, we will evaluate the performance of our model and discuss its potential applications and limitations.

Contents

Chapter 1

Introduction

Face recognition is a technology that enables machines to identify and verify an individual's identity based on their facial features. The technology uses computer algorithms to analyze and compare an individual's face with a database of faces to determine their identity.

Face recognition has become increasingly popular in recent years and has been applied to a wide range of applications, including security and surveillance systems, access control systems, and mobile device authentication. It has also been used for personalization and user experience improvement in fields like entertainment and marketing.

1.1 Face Recognition

Face recognition is a technology that uses computer algorithms to analyze and recognize the unique features of a person's face in an image or video. The technology works by capturing an image or video of a person's face and then using mathematical algorithms to identify and compare specific facial features, such as the distance between the eyes, the shape of the nose and mouth, and the contours of the face.

Face recognition technology can be used for a variety of purposes, such as security and surveillance, access control, and personalization. For example, it can be used to identify individuals in a crowd or to authenticate individuals for access to secure areas. It can also be used for personalization in advertising, by recognizing and targeting specific individuals with tailored messages.

While face recognition technology has many potential benefits, it also raises important ethical and privacy concerns, particularly around the collection and use of personal data. There are concerns about the accuracy and reliability of the technology, as well as the potential for misuse or abuse. Therefore, it is important to carefully consider the potential benefits and risks of face recognition technology and to ensure that it is used in a responsible and ethical manner.

1.2 Various Face Recognition Methods

1.2.1 Face Recognition using GNN

Graph Neural Networks (GNNs) have been successfully applied to many problems related to graph data, including social network analysis, recommendation systems, and molecule property prediction. Face recognition can also be approached using GNNs, particularly when we represent faces as graphs.

One way to represent faces as graphs is to consider each facial landmark (such as eyes, nose, and mouth) as a node in the graph and connect them using edges that represent the spatial relationships between the landmarks. This results in a graph where nodes correspond to facial landmarks, and edges represent the spatial relationships between them. We can then use GNNs to learn embeddings for each node that capture the features of the corresponding landmark.

Once we have learned embeddings for each facial landmark, we can use them to classify faces based on their identity. One approach is to use a fully connected layer on top of the node embeddings to predict the identity of the face. Another approach is to use a Graph Attention Network (GAT) or a Graph Convolutional Network (GCN) to aggregate information from all the nodes and predict the identity of the face.

However, it is worth noting that face recognition using GNNs is still an active research area, and there are many challenges to be addressed. One of the main challenges is to deal with variations in pose, lighting, and facial expression that can affect the spatial relationships between facial landmarks. Additionally, data privacy concerns may limit the amount and quality of the data that can be used to train face recognition models, which can impact their accuracy and performance.

1.2.2 Eigenfaces Method for the Solution of Face Recognition Problem

The basis of the eigenfaces method is the Principal Component Analysis (PCA). Eigenfaces and PCA have been used by Sirovich and Kirby to represent the face images efficiently. They have started with a group of original face images, and calculated the best vector system for image compression. Then Turk and Pentland applied the Eigenfaces to face recognition problem. The Principal Component Analysis is a method of projection to a subspace and is widely used in pattern recognition. An objective of PCA is the replacement of correlated vectors of large dimensions with the uncorrelated vectors of smaller dimensions. Another objective is to calculate a basis for the data set. Main advantages of the PCA are its low sensitivity to noise, the reduction of the requirements of the memory and the capacity, and the increase in the efficiency due to the operation in a space of smaller dimensions. The strategy of the Eigenfaces method consists of extracting the characteristic features on the face and representing the face in question as a linear combination of the so called 'eigenfaces' obtained from the feature extraction process. The principal components of the faces in the training set are calculated. Recognition is achieved using the projection of the face into the space formed by the eigenfaces. A comparison on the basis of the Euclidian distance of the eigenvectors of the eigenfaces and the eigenface of the image under question is made. If this distance is small enough, the person is identified. On the other hand, if the distance is too large, the image is regarded as one that belongs to an individual for which the system has to be trained. The eigenvectors of smaller eigenvalues correspond to smaller variations in the covariance matrix. The discriminating features of the face are retained. The number of eigenvectors depend on the accuracy with which the database is defined and it can be optimized. The group of selected eigenvectors are called the eigenfaces. Once the eigenfaces have been obtained, the images in the database are projected into the eigenface space and the weights of the image in that space are stored. To determine the identity of an image, the eigencoefficients are compared with the eigencoefficients in the database. The eigenface of the image in question is formed. The Euclidian distances between the eigenface of the image and the eigenfaces stored previously are calculated. The person in question is identified as the one whose Euclidian distance is minimum below a threshold value in the eigenface database. If all the calculated Euclidian distances are larger than the threshold, then the image is unrecognizable.

1.2.3 Face Detection Through Template Matching

Template matching is performed first to find the regions of high correlation with the face and eyes templates. Subsequently, using a mask derived from color segmentation and cleaned by texture filtering and various binary operations, the false and repeated hits are removed from the template matching result. The output of this process is then passed to a clustering procedure, where points are within a certain Euclidean distance from one another will be clustered into one point. The whole process will then be repeated at a different scale/resolution. The outputs from each resolution are then recombined into a single mask. For a primary face detection method, we had many possible choices like neural nets, PCA etc. We knew that neural nets are simple to implement but tend to be over-trained on such a limited training sample (especially if we need to split the images into a disjoint test set and training set. Partially occluded faces also may cause a problem, and it is difficult to predict the response of a neural net in these circumstances. PCA involves lowering the

dimensionality of an image and may face the same problem of over-specification of the face-subspace due to the very limited training set. Ultimately, we chose template matching as the primary face detection method due to its conceptual simplicity and our confidence in its inherent extensibility to more general images and predictable response in the case of partially occluded faces, even with such a
limited training set. This effectively pre-empts the problem of obtaining a fair and big sample of training faces from other sources. Another consideration was the ability to combine feature detection with face detection very easily, with only the extra step of creating more templates. Although template matching is basically the two-dimensional cross-correlation of a grayscale image with a grayscale template, hence estimating the degree of similarity
between the two, there are numerous details involved that would have drastic influence on the overall performance of the system.

1.2.4 Convolutional Neural Network for face Recognition

A Convolutional neural network (CNN)is a type of artificial neural network that has one or more convolution layers and are used mainly for image processing, classification, segmentation and also for other auto correlated data. Deep learning is a machine learning based artificial neural network that recognize objects in image by progressively extracting features from data through higher layers. As shown in figure in order to recognize face in an image we have to train the CNN with human faces. The benefit of using CNNs is their ability to develop an internal representation of a two- dimensional image. This allows the model to learn position and scale of faces in an image. After train the CNN it can able to recognize face in an image One can effectively use Convolutional Neural Network for Image data. CNN that extracts features in an image. An image is nothing but the 2-dimensional array. Before training an image, we need to process the dataset. By processing the dataset, we mean converting each image in to NumPy array. Each row represents an image. NumPy package is inbuilt function. Datasets is completely ready to be trained by the model. Neural networks are like layers. Each layer of neural network contains nodes which calculates some values based on characteristics or weights. Activation function are Relu for hidden layers and either sigmoid or SoftMax for output layers. Convolution layer is a fundamental mathematical operation that is highly useful for to detect features of an image. In this layer we pass kernel. i.e., n*n matrix over the image pixel. Kernel has values in each of cell. It processed with original image help to produce some characteristics which help to identify images of the same object while predicting. Max Pooling operation involves sliding a 2- dimensional filter over each channel of features map and extract maximum features from image. Pooling layer used to reduce the dimension of feature map. It reduces the number of parameters to learn and amount of computation to perform.

Chapter 2

Methodology

2.1 Process

To achieve our goal, we must go through the following steps:

- To get facial landmarks from media pipe.
- Use these landmarks to generate signature for each person
- Generate latent representation of unknown person and compare it with all signatures.
- Testing our model with different datasets

2.1.1 Landmark Detection on Face

Media pipe Face Mesh is a pre-built model in the Media pipe framework that provides real-time facial landmark detection, which includes detecting 468 facial landmarks on a human face. The model works by analyzing the 2D geometry of the face, detecting landmarks, and mapping them to a 3D mesh.

The face mesh model in Media pipe is based on a machine learning model that uses a deep neural network trained on a large dataset of labeled images to predict the 3D coordinates of facial landmarks. The model is optimized for real-time performance and can run on a variety of devices, including mobile phones, laptops, and desktop computers.

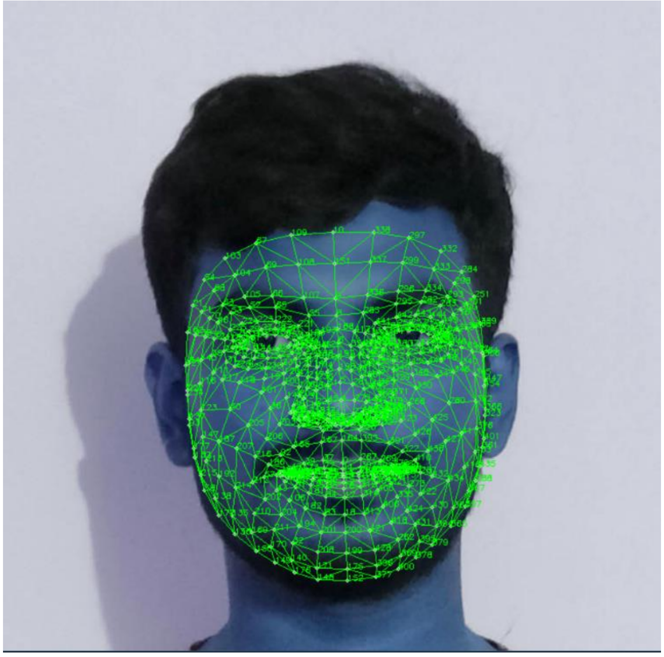


Figure 3.1: Face Landmarks

Using these (468 X 468) landmarks. we are creating two adjacency matrices. first matrix is a binary matrix, which contains information about the edges. Second matrix is also a binary matrix, which contains the information about the edges which is used as a feature matrix in our model.

2.2 Graph Auto Encoder (GAE)

A Graph Auto Encoder (GAE) is a type of neural network architecture designed to learn efficient representations of graph-structured data. GAEs are typically used for tasks such as node embedding and link prediction in graph data.

The GAE architecture consists of two main components: an encoder and a decoder. The encoder takes as input a graph and produces a low-dimensional vector representation (embedding) of each node in the graph. The decoder then takes the node embeddings as input and attempts to reconstruct the original graph structure.

The encoder in a GAE typically consists of multiple graph convolutional layers, which are used to aggregate information from the node's neighbourhood and produce a fixed-length feature vector for each node. The decoder is often a multi-layer perceptron (MLP) that takes the node embeddings and produces a reconstruction of the original graph structure.

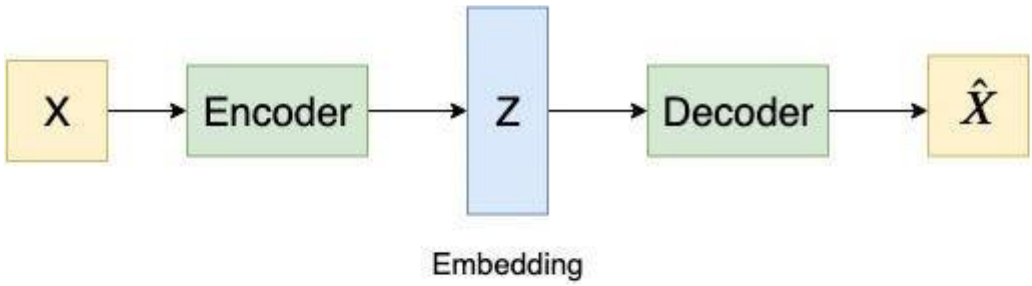
Training a GAE involves optimizing a loss function that measures the difference between the original graph and the reconstructed graph. The loss function typically includes terms to encourage the embeddings to be meaningful and to penalize reconstruction errors.

Overall, GAEs are a powerful tool for learning representations of graph-structured data and can be applied to a variety of tasks in fields such as social network analysis, bioinformatics, and recommendation systems.

The formula for a Graph Auto Encoder (GAE) can be expressed mathematically as follows:

1. Encoder:
   - Input: A graph G with N nodes, represented as an adjacency matrix A and a feature matrix X, where A[i,j] indicates the presence of an edge between node i and node j, and X[i,:] represents the feature vector for node i.
   - Output: A matrix Z of node embeddings, where each row z_i represents the embedding for node i in a low-dimensional space.
   - Calculation: Z = f(A,X), where f is a neural network with multiple graph convolutional layers that aggregates information from the node's neighbourhood and produces a fixed-length feature vector for each node.
2. Decoder:
   - Input: The node embeddings Z obtained from the encoder.
   - Output: A reconstructed adjacency matrix A' that approximates the original adjacency matrix A.
   - Calculation: A' = g(Z), where g is a neural network that takes the node embeddings as input and produces a reconstruction of the original graph structure.
3. Loss function:
   - The loss function measures the difference between the original adjacency matrix A and the reconstructed adjacency matrix A'. A common choice of loss function is the binary cross-entropy loss, which is defined as:

L = -sum (A * log(A') + (1-A) * log(1-A'))
where "*" denotes element-wise multiplication and "sum" denotes the sum of all elements in the matrix. The loss function can also include additional regularization terms to encourage the embeddings to be meaningful and to penalize reconstruction errors. The parameters of the GAE (i.e., the weights and biases of the neural networks) are trained by minimizing the loss function using an optimization algorithm such as stochastic gradient descent.



2.3 Edge Weight in GAE

In the context of reinforcement learning, the Generalized Advantage Estimation (GAE) algorithm is used to estimate the advantage function, which is a measure of how much better an action is compared to the average action in a particular state.

The edge weight in GAE refers to the weight or importance given to the transition between states during the estimation of the advantage function. In the case of edge weights, the immediate reward and expected future reward are multiplied by the edge weight, which allows for assigning different levels of importance to different transitions.

The GAE algorithm can incorporate edge weights in several ways. One approach is to use the edge weight to modify the temporal difference error δ(t+i) term in the GAE formula, which can be expressed as:

δ_weighted(t+i) = w(t+i) * δ(t+i)

where w(t+i) is the edge weight for the transition from state t+i to t+i+1.

Alternatively, the edge weight can be incorporated directly into the GAE formula as follows:

GAE_weighted(t) = ∑i=0∞ (γλ)^i * w(t+i) * δ(t+i)

where w(t+i) is the edge weight for the transition from state t+i to t+i+1.

The edge weight in GAE can be used to assign more importance to certain state transitions based on factors such as their perceived relevance, reliability, or difficulty. The weighted GAE algorithm can thus provide a more fine-grained estimation of the advantage function, leading to improved performance in reinforcement learning tasks.

$$
\begin{aligned}
\text{GGNN: } A' &= GRU\big(A \,,\; \boldsymbol{W}_1 \cdot B + \boldsymbol{W}_2 \cdot C + \boldsymbol{W}_1 \cdot D \big) \\
\boxed{\text{R-GCN: } A' = \sigma\big(\boldsymbol{W}_{\circlearrowleft} \cdot A + \boldsymbol{W}_1 \cdot B + \boldsymbol{W}_2 \cdot C + \boldsymbol{W}_1 \cdot D \big)} \\
\text{R-GAT: } A' &= \sigma\big((\boldsymbol{a}_{A'})_{A \xrightarrow{\circlearrowleft} A} \cdot \boldsymbol{W}_{\circlearrowleft} \cdot A + (\boldsymbol{a}_{A'})_{B \xrightarrow{1} A} \cdot \boldsymbol{W}_1 \cdot B + (\boldsymbol{a}_{A'})_{C \xrightarrow{2} A} \cdot \boldsymbol{W}_2 \cdot C + (\boldsymbol{a}_{A'})_{D \xrightarrow{1} A} \cdot \boldsymbol{W}_1 \cdot D \big) \\
\text{R-GIN: } A' &= \sigma\big(MLP_{\circlearrowleft}(A) + MLP_1(B) + MLP_2(C) + MLP_1(D) \big) \\
\text{GNN-MLP: } A' &= \sigma\big(MLP_{\circlearrowleft}(A \| A) + MLP_1(B \| A) + MLP_2(C \| A) + MLP_1(D \| A) \big) \\
\text{RGDCN: } A' &= \sigma\big(\boldsymbol{W}_{\circlearrowleft, A} \cdot A + \boldsymbol{W}_{1,A} \cdot B + \boldsymbol{W}_{2,A} \cdot C + \boldsymbol{W}_{1,A} \cdot D \big) \\
\text{GNN-FiLM: } A' &= \sigma\big(\boldsymbol{\beta}_{\circlearrowleft, A} + \boldsymbol{\gamma}_{\circlearrowleft, A} \odot \boldsymbol{W}_{\circlearrowleft} \cdot A + \boldsymbol{\beta}_{1,A} + \boldsymbol{\gamma}_{1,A} \odot \boldsymbol{W}_1 \cdot B + \boldsymbol{\beta}_{2,A} + \boldsymbol{\gamma}_{2,A} \odot \boldsymbol{W}_2 \cdot C + \boldsymbol{\beta}_{1,A} + \boldsymbol{\gamma}_{1,A} \odot \boldsymbol{W}_1 \cdot D \big)
\end{aligned}
$$

Figure 4.3: Some methods of adding edge weight in GAE

2.4 Recognition and Comparison

For comparing two images, we are finding similarity between latent representations of two images. We are finding Cosine distance, Euclidian distance, and Pearson distance.

We are calculating similarity of latent representation of unknown image with all the signatures which are present in our database.

Euclidian Distance: Euclidean distance is a mathematical concept used to measure the distance between two points in a multi-dimensional space. It is also known as the "straight-line distance" or the "as-the-crow-flies" distance.

In a two-dimensional space, the Euclidean distance between two points, say (x1, y1) and (x2, y2), can be calculated using the Pythagorean theorem as:

distance = sqrt((x2-x1)^2 + (y2-y1)^2)

Cosine Distance: Cosine distance is a measure of similarity between two vectors in a multi-dimensional space. It is commonly used in machine learning and natural language processing to compare the similarity of documents or words represented as vectors.

The cosine distance between two vectors A and B is calculated as the cosine of the angle between them. Mathematically, it is defined as:

Cosine distance (A, B) = 1 - (A . B) / (||A|| * ||B||)

where A.B is the dot product of vectors A and B, and ||A|| and ||B|| are the magnitudes of vectors A and B, respectively.

Pearson Distance: Pearson distance, also known as Pearson correlation coefficient, is a measure of the linear correlation between two variables in a dataset. It is often used in statistical analysis and machine learning to identify the relationship between variables.

The Pearson distance between two variables X and Y is calculated as the covariance of X and Y divided by the product of their standard deviations. Mathematically, it is defined as:

pearson_distance(X, Y) = cov(X, Y) / (std(X) * std(Y)) where cov(X, Y) is the covariance of X and Y, and std(X) and std(Y) are the standard deviations of X and Y, respectively.

The Pearson distance ranges from -1 to 1, with 1 indicating a perfect positive correlation between the two variables, 0 indicating no correlation, and -1 indicating a perfect negative correlation. Therefore, a higher Pearson distance implies a stronger positive correlation between the two variables, while a lower Pearson distance implies a stronger negative correlation between them.

Chapter 3

Experiment and Result

Dataset

- In this project we have used Yale and Orl dataset which has 165 grayscale images, 11 different images for each of the 15 subjects.
- This image has both expression and light variation with face occlusion like glasses on few images.

To test our model, we have divided the dataset into two different parts

- Neutral face image (30 images)
- Non-neutral face image (135 images)

Neutral faces: Faces having no expression, with good lighting conditions and images in which full face is visible

Non-Neutral faces: if subject is showing some type of expression, or image is not captured properly, full face is not visible. Then our model can't recognize these types of images.

Results Table:

In our results table, we included results of experiment done in different scenarios. The header of the table contains the information's like, Dataset name, number of signatures, difficulty-level, edge features are used or not, and the results. I Easy difficulty level, we used neutral faces, and in Difficult level, we used non-neutral faces

Experiment and Results

Experiment number	Difficulty Level	Number of Signatures for each person	Number of Test Images of each person	Dataset	Edge Feature	Result
1	Easy	1	1	Orl	Yes	90%
2	Easy	1	1	Orl	No	80%
3	Difficult	1	9	Orl	Yes	51%
4	Difficult	1	9	Orl	No	45%
5	Easy	1	1	Yale	Yes	100%
6	Easy	1	1	Yale	No	93%
7	Difficult	1	10	Yale	Yes	50%
8	Difficult	1	10	yale	No	46%

Chapter 4

Conclusion

Face recognition ML models are becoming increasingly popular due to their ability to accurately identify individuals from images or videos. This model uses GNN to extract facial features and create a unique representation of each individual's face, which can then be used for identification purposes.

This work aims to recognize persons using images of their faces. There has been a significant progress in the Face Recognition

With edge features with Graph Auto Encoder (GAE) there has been a significant improvement in the accuracy, which has led to the development of specialized models.

4.1 Limitations

With this project is showing nice accuracy for neutral faces (Faces having no expression, with good lighting conditions and images in which full face is visible). But for non-neutral faces, our model is showing only 50% accuracy. Which means, if subject is showing some type of expression, or image is not captured properly, full face is not visible. Then our model can't recognize these types of images.

4.2 Further work

4.2.1 Create a GUI for face recognition.

4.2.2 Increase accuracy for non-neutral faces.

4.2.3 Integrating features for recognizing subjects using images of there partial faces

4.2.4 Increase the overall accuracy.

We took up the objective to increase our accuracy for non-neutral faces and to create GUI for face recognition, so we will continue to develop and improvise it.

Bibliography

[1] Lawrence, Steve, et al. "Face recognition: A convolutional neural-network approach." *IEEE transactions on neural networks* 8.1 (1997): 98-113.
[2] Lawrence, S., Giles, C. L., Tsoi, A. C., & Back, A. D. (1997). Face recognition: A convolutional neural-network approach. *IEEE transactions on neural networks*, *8*(1), 98-113.
[3] Lawrence, Steve, C. Lee Giles, Ah Chung Tsoi, and Andrew D. Back. "Face recognition: A convolutional neural-network approach." *IEEE transactions on neural networks* 8, no. 1 (1997): 98-113.
[4] Allamanis, M., Brockschmidt, M., and Khademi, M. Learning to represent programs with graphs. In International Conference on Learning Representations (ICLR), 2018.
[5] Ba, L. J., Kiros, R., and Hinton, G. E. Layer normalization. CoRR, abs/1607.06450, 2016.
[6] Balog, M., van Merrienboer, B., Moitra, S., Li, Y., and Tar- ¨ low, D. Fast training of sparse graph neural networks on dense hardware. CoRR, abs/1906.11786, 2019.
[7] Busbridge, D., Sherburn, D., Cavallo, P., and Hammerla, N. Y. Relational graph attention networks. CoRR, abs/1904.05811, 2019.