**A**

**Major Project on**

# Deep Relation

**BACHELOR OF ENGINEERING**

**In**

**COMPUTER SCIENCE AND ENGINEERING**

**Submitted by**

|  |  |
| --- | --- |
| **Modala Shyam Prasad :** | **2453-21-733-103** |
| **Gande Bhanu Tharan :** | **2453-21-733-081** |
| **Jagati Priya Ragini :** | **2453-21-733-086** |
| **Muppidi Naga Manogna :** | **2453-21-733-106** |
| **Kokkalla Sai Poojitha :** | **2453-21-748-032** |
| **Panaganti Jyothirmai :** | **2453-21-748-042** |

**Under the Guidance**

**of**

**Dr. Shilpa Choudhary**

**(Asst. Professor)**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING Neil Gogte Institute of Technology**

Kachivani Singaram Village, Hyderabad, Telangana 500058.

**February,2024**

# CERTIFICATE

This is to certify that the Project Report (Project work - II) Titled **“DEEP**

**RELATION”** is being submitted by **Modala Shyam Prasad (2453-21-733-**

## 103), Gande Bhanu Tharan (2453-21-733-081), Jagati Priya Ragini

**(2453-21-733-086), Muppidi Naga Manogna (2453-21-733-106), Kokkalla Sai Poojitha (2453-21-748-032), Panaganti Jyothirmai (2453-21-748-042)** in **III B.E.– V Semester** of **Computer Science And Engineering** is a record bonafide work carried out by them.

**Head Of the Dept**

**Internal Guide**

# DECLARATION

We the students of Neil Gogte Institute of Technology hereby declare that the Project work Report entitled, **“Deep Relation”** submitted during V semester for the B.E. degree is entirely our work and all ideas and references have been duly acknowledged.

**Submitted By:**

**Modala Shyam prasad**

**(2453-21-733-103)**

**Gande Bhanu Tharan**

**(2453-21-733-081)**

**Jagati Priya Ragini**

**(2453-21-733-086)**

**Muppidi Naga Manogna**

**(2453-21-733-106)**

**Kokkalla Sai Poojitha**

**(2453-21-748-032)**

**Panaganti Jyothirmai**

**(2453-21-748-042)**

# ACKNOWLEDGEMENT

We are happy to express our deep sense of gratitude to the principal of the college **Dr. R Shyam Sunder**, Professor, Neil Gogte Institute of Technology, for having provided us with adequate facilities to pursue our project.

We would like to thank **Prof. K V Ranga Rao**, Professor and Head,

Department of Computer Science and Engineering, Neil Gogte Institute of Technology, for having provided the freedom to use all the facilities available in the department, especially the laboratories and the library.

We are very grateful to our project guide **Dr. Shilpa Choudhary**, Assistant Professor, Department of Computer Science and Engineering, Neil Gogte Institute of Technology, for her extensive patience and guidance throughout our project work.

We sincerely thank our seniors and all the teaching and non-teaching staff of the Department of Computer Science & Engineering for their timely suggestions, healthy criticism and motivation during the course of this work.

We would also like to thank classmates for always being there whenever we needed help or moral support. With great respect and obedience, we thank our parents, sisters and brother who were the backbone behind our deeds.

Finally, we express our immense gratitude with pleasure to the other individuals who have either directly or indirectly contributed to our need at the right time for the development and success of this work

## ABSTRACT

The "Deep Relation" project is a machine learning application designed to determine the relationship between two inputs using a Siamese network model. The Siamese network architecture is employed to learn and encode the features of the input data into a shared space, where a similarity metric can be computed to assess the relationship between them.

The project involves training the Siamese network on pairs of inputs along with their corresponding labels indicating the relationship between them (e.g., related or unrelated). Through the process of training, the model learns to distinguish between related and unrelated pairs by optimizing a similarity function that minimizes the distance between similar pairs and maximizes the distance between dissimilar pairs in the shared feature space.

Once trained, the Deep Relation model can efficiently assess the relationship between new pairs of inputs by computing their similarity scores. This capability makes it suitable for various applications such as similarity-based search, recommendation systems, and information retrieval tasks.

The documentation provides detailed information on the architecture of the Siamese network, the training process, input data preparation, model evaluation metrics, and guidelines for deploying and using the trained model in real-world scenarios.

# CONTENTS

S.NO. TITLE PAGE NO.

## CERTIFICATION ii

## DECLARATION iii

## ACKNOWLEDGMENT iv

## ABSTRACT v

## TABLE OF CONTENTS vi

## LIST OF FIGURES vii

## INTRODUCTION

## 1.1 Problem Statement 10

## 1.2 Solution 11

## 2 LITERATURE SURVEY

## 2.1 Surveys 12

## 2.2 Existing System 13

## 2.3 Disadvantages of Existing System 13

**3** PROPOSED SYSTEM

3.1 WorkFlow 14

3.2 Proposed System Advantages 15

3.3 Requirements 15

3.4 Data Collection 16

3.5 Data Preprocessing 16

3.6 Model Building 16

3.7 Siamese Network Architecture 17

3.8 Training The Model 18

3.9 Saving he Model 20

3.10 Testing The Model 20

**4. RESULT**

4.1 Graphs 22

**5. CONCLUSION**

5.1 Future Plan 36

**6. REFERENCES** 38

# LIST OF FIGURES

3.1.1 Work Flow of Proposed System 14

3.5.1 Images from Dataset After Data Augmentation 16

3.7.1 Siamese Network Architecture 17

# 3.8.1Different hyperparameters used in model building 19

3.10.1 Accuracies with different activation functions 21

**GRAPHS**

3 Layers:

4.1.1.Relu with Adam 24

4.1.2.Relu with SGD 25

5 Layers :

4.1.3.Relu with Adam 26

4.1.4 Relu with SGD 27

4.1.5 P.elu with Adam 28

4.1.6 P.elu with SGD 29

4.1.7.Gelu with Adam 30

4.1.8 Gelu with SGD 31

4.1.9 .elu with Adam 32

4.1.10 elu with SGD 33

7 Layers:

4.1.11.Relu with Adam 34

4.1.12 Relu with SGD 35

# 

# INTRODUCTION

Introducing "Kinship relation," a groundbreaking platform revolutionizing the exploration of blood relationships through cutting-edge deep learning techniques, including sophisticated Siamese networks. In today's interconnected world, understanding familial connections holds immense importance, yet traditional methods often prove cumbersome and inconclusive. Kinship relation addresses this challenge head-on, offering users an intuitive and efficient solution to unravel the complexities of genetic ties.

Through its intuitive interface and robust functionality, Kinship relation streamlines the process of investigating genetic relationships. By harnessing the power of deep learning convolutional layers and specialized Siamese networks, Kinship relation examine facial features with unparalleled accuracy, providing users with comprehensive insights into their genetic connections. These Siamese networks enable Kinship relation to compare and evaluate genetic similarities between pairs of individuals, enhancing the precision of its assessments.

Whether users are delving into their ancestry or navigating complex medical histories, Kinship relation empowers them with actionable information tailored to their unique genetic profiles. With a commitment to precision and user empowerment, Kinship relation sets a new standard in the realm of blood relationship discovery. By leveraging advanced technology, including state-of-the-art Siamese networks, to decode the intricacies of genetic connections, Kinship relation offers a transformative approach to understanding familial relationships.

## 1.1 PROBLEM STATEMENT

Identifying blood-related individuals using deep learning techniques presents a significant challenge across diverse fields, including genealogy, forensic science, and medical genetics. Current methods often lack the precision and efficiency required to determine genetic relationships accurately between two given individuals. This creates a pressing need for a reliable and scalable solution capable of discerning blood ties with confidence, especially in scenarios where genetic information is pivotal for personal, legal, or medical purposes. Hence, the primary problem revolves around devising an accurate method to ascertain whether two given individuals share a blood relationship, leveraging the capabilities of deep learning technologies to address this challenge effectively.

## 1.2 SOLUTION

To address the challenge of identifying blood-related individuals, we propose leveraging Siamese networks, specialized neural network architectures recognized for their efficacy in comparing and evaluating similarities between pairs of data points. Our approach involves training the Siamese network on a dataset containing pairs of known blood-related and unrelated individuals. During inference, the Siamese network computes a similarity score between the facial embeddings or other relevant genetic markers of the two input individuals, providing a quantitative measure of their genetic relatedness. Through rigorous experimentation and validation, we ensure that our solution achieves high levels of accuracy and reliability in discerning genetic ties, thereby offering a robust framework for blood relationship identification using deep learning techniques. By addressing the limitations of traditional methods and leveraging the power of Siamese networks, our solution provides a promising avenue for accurately identifying blood-related individuals across various applications.

# 2.LITERATURE SURVEY

## 2.1 SURVEYS

1. Fang *et al.* [[1](https://ietresearch.onlinelibrary.wiley.com/doi/full/10.1049/iet-ipr.2018.5552#ipr2bf01700-bib-0004)] were among the first researchers who tackled the kinship verification. They used the pictorial structures model to locate facial parts (eyes, nose, mouth etc.); then they extracted a set of 22 low-level features from these parts including, parts colour and geometry, distances between parts and gradient of the face. The *K*-nearest-neighbour and support vector machine (SVM) are used to classify the pairs of images as having kin relation or not.
2. Xiaoting Wu *et al.* [[2](https://ietresearch.onlinelibrary.wiley.com/doi/full/10.1049/iet-ipr.2018.5552#ipr2bf01700-bib-0009)] investigated the usefulness of colour information in the kinship verification from facial images. To encode both the chrominance and the luminance information in the colour images, they extracted joint colour–texture features. The performance of kinship verification using joint colour–texture [LPQ, binarised statistical image feature (BSIF), and neighbourhood repulsed metric learning (NRML)] in the three colour spaces (RGB, HSV, and YCbCr) is then compared against approaches using only grey-scale information.
3. Lu *et al.* [[3](https://ietresearch.onlinelibrary.wiley.com/doi/full/10.1049/iet-ipr.2018.5552#ipr2bf01700-bib-0014)] learned that the distances between pair image in same classes are as small as possible, while the distances are as large as possible between pair image different class based on textural features.
4. Fang *et al.* [[4](https://ietresearch.onlinelibrary.wiley.com/doi/full/10.1049/iet-ipr.2018.5552#ipr2bf01700-bib-0014)] learned similarity matrix for kinship verification, by proposing a sparse similarity metric learning (SSML) method which enforces both the PSD constraints and the group sparsity.
5. Hu *et al.* [[5](https://ietresearch.onlinelibrary.wiley.com/doi/full/10.1049/iet-ipr.2018.5552#ipr2bf01700-bib-0015)] learned multiple global and local distance metrics, by maximising the correlations of features and the distance between each positive pair is less than a low threshold, and that for each negative pair is greater than a high threshold.
6. Zhang *et al.* [[6](https://ietresearch.onlinelibrary.wiley.com/doi/full/10.1049/iet-ipr.2018.5552#ipr2bf01700-bib-0014)] proposed deep convolutional neural networks (CNNs) to extract the high-level features which were produced from the neuron activations of the last hidden layer, and then fed the extracted feature into soft-max classifier to verify the kinship relationships. They also extracted the key-points-based features.
7. Yan *et al.* [[7](https://ietresearch.onlinelibrary.wiley.com/doi/full/10.1049/iet-ipr.2018.5552#ipr2bf01700-bib-0014)] learned a set of hierarchical non-linear transformations based on various features by training a deep neural network, then reduced the distance of the positive pair also, that enlarged the distance of negative pair.

8 .Boutella *et al.* [[8](https://ietresearch.onlinelibrary.wiley.com/doi/full/10.1049/iet-ipr.2018.5552#ipr2bf01700-bib-0014)] proposed a method to address the kinship verification challenge from spatiotemporal descriptors and explore the use of texture features (BSIFTOP, LPQTOP, and LBPTOP) and deep features (CNN) for characterising faces; his experiments based on videos over still images. They have performed various experiments on UvA-NEMO Smile database. SVM was used to classify a pair of facial features as a positive or negative sample.

## 2.2 EXISITING SYSTEM

While existing systems for kinship relation identification offer valuable tools and insights, they also have certain limitations that may have prompted the exploration of alternative approaches such as Siamese network models.

## 2.3 DISADVANTAGES

1. Limited Genetic Data

2. Privacy Concerns

3. Database Fragmentation

4. Complex Interpretation

5. Inaccurate Matches

# 3.PROPOSED SYSTEM

**3.1 WORKFLOW**

## 

## Fig 3.1.1 Workflow of proposed System

**3.2 PROPOSED SYSTEM ADVANTAGES**

Siamese networks excel in comparing and analyzing similarities between pairs of data points, making them ideal for tasks like kinship verification. By leveraging shared weights and learning discriminative features, Siamese networks offer a powerful approach to accurately determining genetic relatedness between individuals.The advantages are given below.

1. Effective Handling of Complex Relationships

2. Scalability

3. Transfer Learning

4. End-to-End Training

* 1. **REQUIREMENTS**

**Hardware Requirements :** T4 GPU

**Software Requirements :** Google Colab

**Technologies Used :** Pytorch**,** Deep learning **,** Siamese Network**,** Convolutional Neural Networks

## 3.4 Data Collection

The "Families in the Wild (FIW)" dataset was used.11,932 real family photos from 1000 families make up FIW. It is composed of 11 relationships and 656,954 image pair. This dataset serves as the primary source for this project due to its complexity and suitability for the deep learning idea. This dataset includes pictures of various families. First, the dataset includes several family folders, each of which has folders with images of the family members taken from various perspectives.

## 3.5 Data Preprocessing

However, some improvement is necessary to achieve better outcomes, and this also aids in model construction so that data pre-processing is used. We used data augmentation during the pre-processing stage of the data to expand the dataset in order to train the model. We made the following modifications to the dataset as part of our additional updates, which included data augmentation, Flip both vertically and horizontally, turning round, Blurring of Colour, Modified cropping size, the use of grayscale.



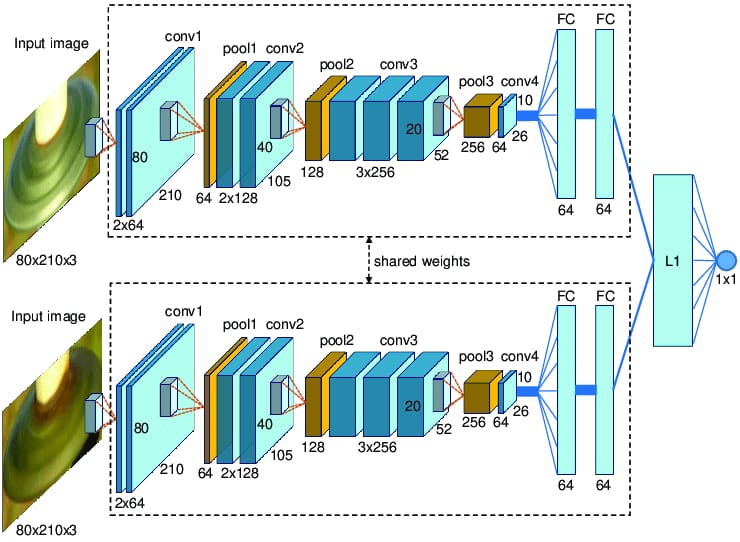
**Fig 3.5.1** Images from dataset after Data Agumentation

## 3.6 Model Building

In our project, the Siamese network model is employed. Usually, two identical subnetworks, or branches, with the same weights and parameters make up a Siamese network architecture. One input is processed independently by each branch. After that, the outputs from the two branches are combined and contrasted using a similarity metric like cosine similarity or Euclidean distance. In order to minimise the distance between similar pairs and maximise the distance between dissimilar pairs, the network is trained using pairs of similar and dissimilar inputs. For tasks like similarity learning, where determining the degree of similarity or dissimilarity between pairs of data points is the objective, Siamese networks are frequently utilised.

**3.7 Siamese Network Architecture**

The architecture of the Siamese network is represented by a figure in the below figure. This is a thorough explanation of the architecture and how it functions. Initially, two images are captured and divided into two sub-branch.The first sub-branch processes the input images using convolution layers, followed by max pooling, flattening, and formation of dense layers. Here, we experimented with a variety of optimizers and activation functions to determine which would be best for the project in terms of accuracy. In order to lower the loss function, we also employed varying numbers of convolution layers with various activation functions and optimizers .

 **Fig 3.7.1** Siamese Network Architecture

**3.8 Training the model**

within the siamese network Initially, convolution layers are formed by taking images. To build the best model, this process is repeated using varying numbers of convolution layers, such as three, five, and seven, then each convolution layer's maxpooling is completed. By lowering the spatial dimensions to the point where only the most crucial information is retained, the max pooling concept helps to facilitate feature extraction and lower the computational complexity of the network. Convolutional neural networks (CNNs) employ the padding technique to regulate the spatial dimensions of the feature maps generated by their convolutional layers. Before doing the convolution operation, more pixels—typically zeros—are added around the input data. Subsequent flattening layers are created once the activation function is applied to it; here, Gelu, Relu, PRelu,Elu, swish and sigmoid were utilised.

The flattening layers are made after every convolution has been made. Next, backpropagation was carried out after about 30 iterations to ensure that the loss function was as little as possible. An optimizer is used to minimise the loss function during backpropagation. Adam and SGD are the optimizers that we tried on in this case.In addition, batch size, dropout ratio, and seeding are employed as factors.These additional hyperparameters are utilised to improve the model and produce better outcomes. Setting the random number generator's seed to a fixed value prior to conducting an experiment or training a model is referred to as "seeding" in the context of machine learning and deep learning. The number of training samples used in a single optimisation algorithm iteration is referred to as the batch size. The dataset is split up into smaller subsets, or batches, for model training, and the network processes each batch in a single iteration. Dropout is a regularisation strategy in deep learning that involves randomly deleting (zeroing out) a percentage of a layer's neurons during training in order to prevent overfitting.The information about the hypermeters used is mentioned in the below table.

# Table 3.8.1 Different hyperparameters used in model building

|  |  |
| --- | --- |
| **Name of the hyperparameter** | **Types used** |
| Number of convolution layers | Three |
| Five |
| Seven |
| Types of Optimizers | Adam |
| SGD (Stochastic Gradient Descent) |
| Types of Activation functions | Relu |
| PRelu |
| Gelu |
| Swish |
| Sigmoid |
| Elu |
| pooling | Maxpooling |
| Seeding | 42 |
| Batch Size | 32 |
| Epochs | 30 |
| Dropout ratio | 0.3 |

## 3.9 Saving the model

The model is saved for testing after it has been trained, depending on how many iterations it took .Following testing of the model, the outcomes are compared to the predictions.

**3.10 Testing the model**

We have 6282 testing images in our dataset. A sample\_submissions.csv file is taken for testing the model where the file contains image pairs which are initially taken as non-related. By giving that images to the model for testing it will check weather they are related or not , if the image pairs are related it will change it from 0 to 1.We had also created our own csv file called celeb.csv file which contains real world images and testing with our model and predicting the results.

|  |  |  |
| --- | --- | --- |
| **Activation Function** | **Optimizers** | **Accuracies** |
| ReLU | SGD | 61.82% |
| ADAM | 63.64% |
| PReLU | SGD | 62.04% |
| ADAM | 58.14% |
| GELU | SGD | 61.32% |
| ADAM | 59.12% |
| SWISH | SGD |  |
| ADAM | 58.45% |
| SIGMOID | SGD | 55.07% |
| ADAM | 59.63% |
| ELU | SGD |  |
| ADAM | 61.49% |
| SOFTMAX | SGD |  |
| ADAM | 49.32% |

**Table 3.10.1**  Accuracies with different activation functions in Siamese network

**4. RESULT**

## 4.1 Graph

## ACCURACY

Accuracy is a metric that measures the overall correctness of predictions, representing the ratio of correctly predicted instances to the total number of instances in a dataset. It is a common evaluation measure for classification models but may not be suitable for imbalanced datasets.

**Accuracy=(TP+TN)/(TP+TN+FP+FN)**

## PRECISION

Precision is a metric in machine learning that quantifies the accuracy of positive predictions made by a model. It is the ratio of true positives (correctly predicted positive instances) to the sum of true positives and false positives (incorrectly predicted as positive instances). Precision focuses on the quality of positive predictions, emphasizing the model's ability to avoid false positives.

**Precision = TP/(TP+FP)**

## RECALL

Recall, also known as sensitivity or true positive rate, is a metric in machine learning that measures the ability of a model to capture all relevant instances of a class. It is the ratio of true positives (correctly predicted positive instances) to the sum of true positives and false negatives (instances of the class that were not predicted as positive). Recall emphasizes the model's capability to avoid missing positive instances, making it particularly important in scenarios where identifying all relevant cases is crucial.

**Recall=TP/(TP+FN)**

## F1-SCORE

The F1-score is a metric in machine learning that combines precision and recall into a single measure, providing a balance between the two. It is the harmonic mean of precision and recall, calculated as 2 times the product of precision and recall divided by the sum of precision and recall. The F1-score is especially useful when there is an uneven class distribution or when both false positives and false negatives are important considerations, offering a comprehensive evaluation of a model's performance.

**F1 Score = (2\*Precision \*Recall)/(Precision+Recall)**

In the above graphs ,

‘Blue’ coloured line indicates ‘LOSS’

‘Green’ coloured line indicates ‘ACCURACY’

‘Yellow’ coloured line indicates ‘PRECISION’

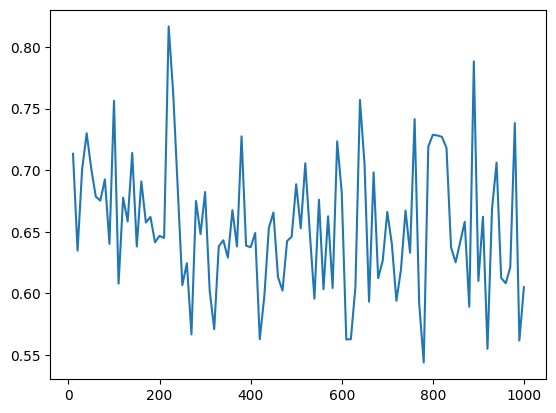
‘Red’ coloured line indicates ‘RECALL’

‘Violet’ coloured line indicates ‘F1-SCORE’

## 3 Layers

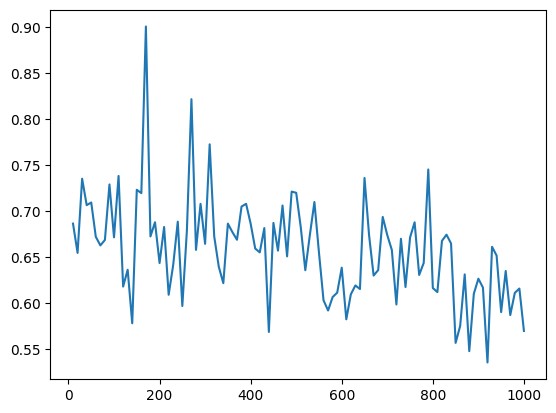
These are the Graphs between Loss and Iterations . Here One Epoch includes 100 iterations.

## RELU with ADAM:



**Fig 4.1.1** Relu with adam

* **Loss:** It appears the loss starts high and trends downwards over the iterations, which is a **positive** sign. This suggests the model's performance is **improving** as it is being trained.



**Fig 4.1.2** Relu with SGD

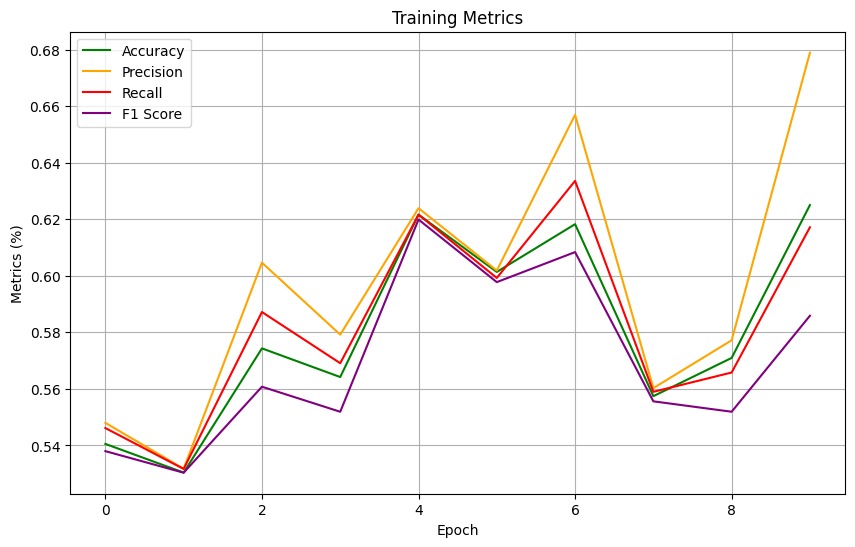
* **Loss:** It appears the loss starts high and trends downwards over the iterations, which is a **positive** sign. This suggests the model's performance is **improving** as it is being trained.

## 5 Layers

The graph shows the training metrics of a model, which includes accuracy, precision, recall, and F1 score. These metrics are plotted against epochs, which are iterations of the training process.

## RELU :

## ADAM:



**Fig 4.1.3** Relu with adam

**Accuracy:** In the graph, it starts at around 0.54 and increases to 0.68 over the epochs.

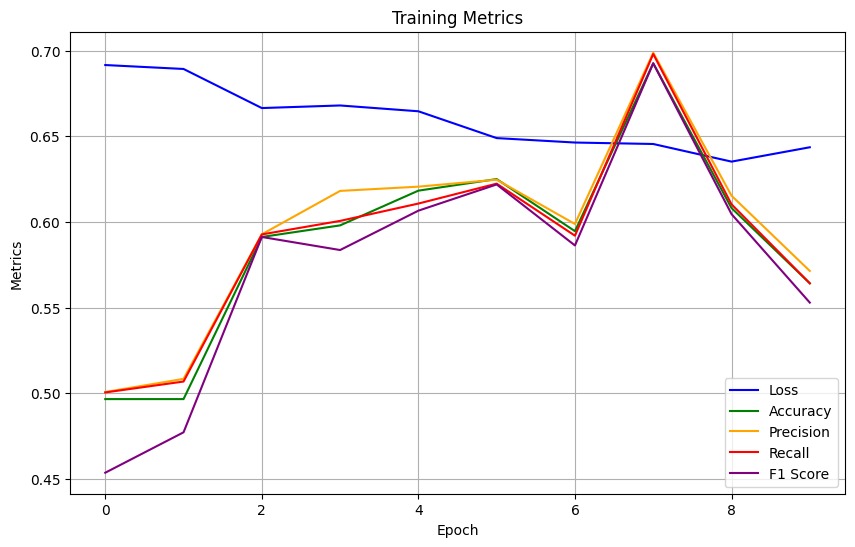
**Precision:** It starts at around 0.56 and increases to 0.66 over the epochs.

**Recall:** It starts at around 0.54 and increases to 0.64 over the epochs.

**F1 Score:** It starts at around 0.55 and increases to 0.65 over the epochs.

## 

## SGD:



**Fig 4.1.4** Relu with SGD

**Loss: In the graph, loss starts around 0.69 and decreases to 0.64 over the epochs.**

**Accuracy:** In the graph, it starts at around 0.50 and increases to 0.70 over the epochs.

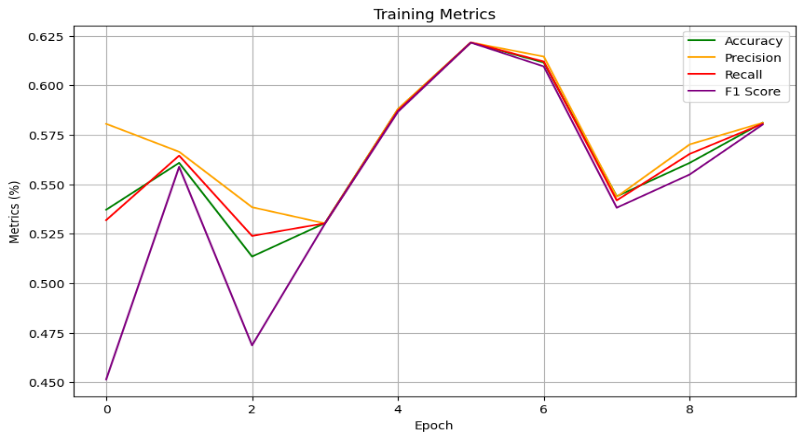
**Precision:** It starts at around 0.52 and increases to 0.62 over the epochs.

**Recall:** It starts at around 0.51 and increases to 0.61 over the epochs.

**F1 Score:** It starts at around 0.51 and increases to 0.60 over the epochs.

## PRELU :

## ADAM:



**Fig 4.1.5** PRelu with adam

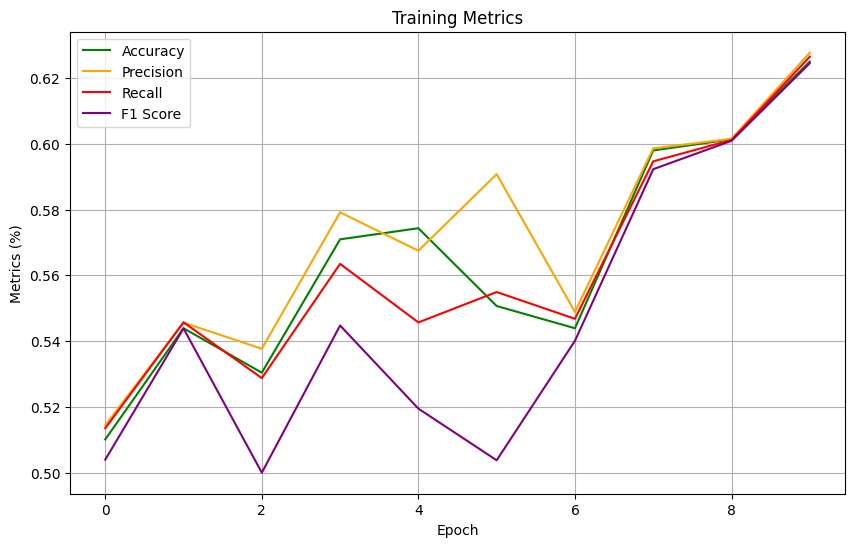
**Accuracy:**  In the graph, it starts at around 0.475 and increases to 0.625 over the epochs.

**Precision:**   It starts at around 0.50 and increases to 0.60 over the epochs.

**Recall:** It starts at around 0.475 and increases to 0.575 over the epochs.

**F1 Score:** It starts at around 0.450 and increases to 0.580 over the epochs.

## SGD:



**Fig 4.1.6** PRelu with SGD

**Accuracy:**  In the graph, it starts at around 0.54 and increases to 0.68 over the epochs.

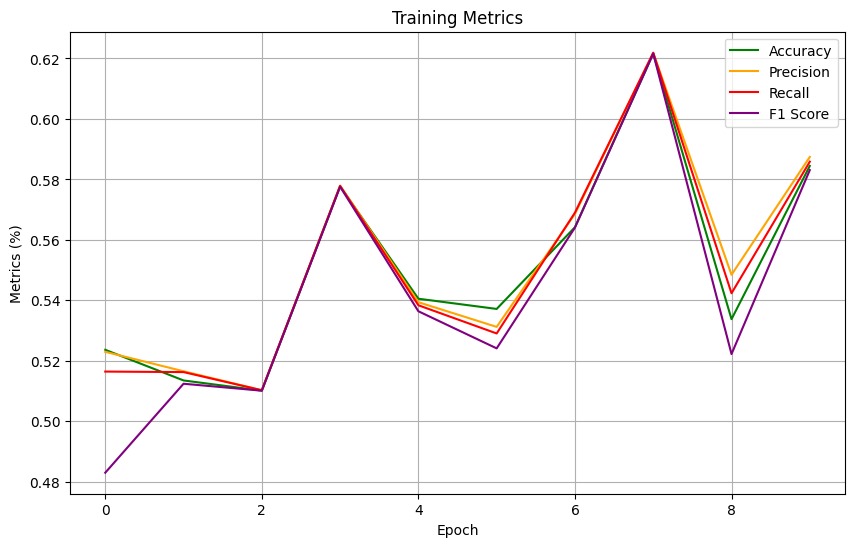
**Precision:** It starts at around 0.56 and increases to 0.66 over the epochs.

**Recall:**  It starts at around 0.54 and increases to 0.64 over the epochs.

**F1 Score:** It starts at around 0.55 and increases to 0.65 over the epochs.

## GELU :

## ADAM:



**Fig 4.1.7** Gelu with adam

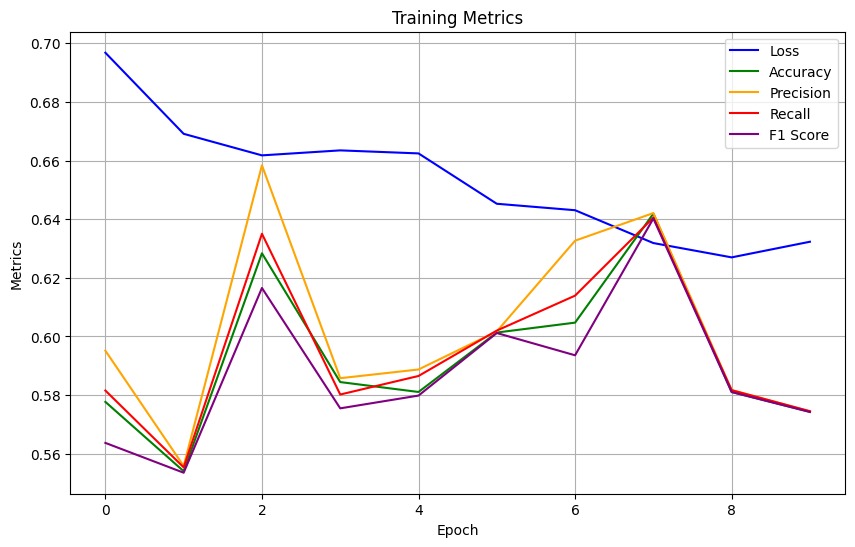
**Accuracy:**  In the graph, it starts at around 0.54 and increases to 0.68 over the epochs.

**Precision:**  It starts at around 0.56 and increases to 0.66 over the epochs.

**Recall:** It starts at around 0.54 and increases to 0.64 over the epochs.

**F1 Score:** It starts at around 0.48 and increases to 0.585 over the epochs.

## SGD:



**Fig 4.1.8** Gelu with SGD

**Loss: In the graph, loss starts around 0.69 and decreases to 0.63 over the epochs.**

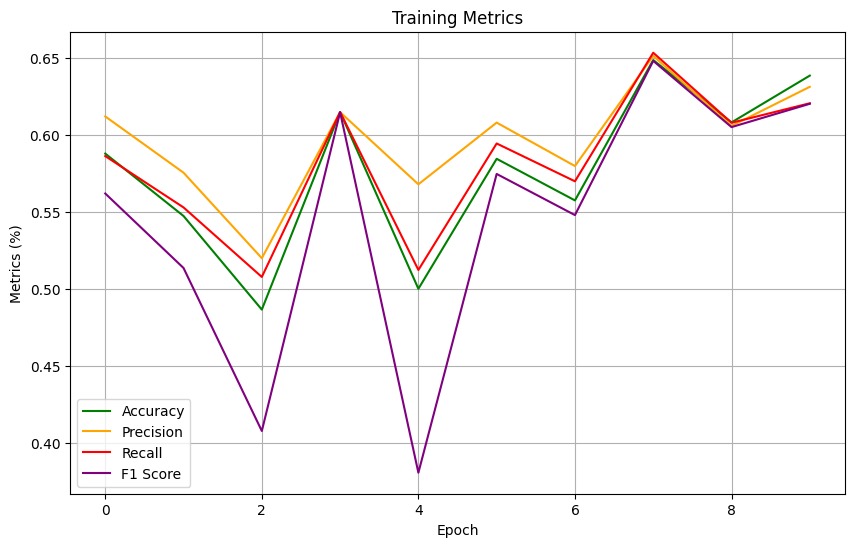
**Accuracy:** is increasing, which means the model is getting better at making correct predictions. It starts at around 0.54 and increases to 0.68 over the epochs.

**Precision:** is also increasing, which means the model is getting better at identifying true positives. It starts at around 0.56 and increases to 0.66 over the epochs.

**Recall:** is slightly increasing, but the increase is not as significant as the other metrics. It starts at around 0.54 and increases to 0.64 over the epochs.

## ELU :

## ADAM:



**Fig 4.1.9** Elu with adam

* **Accuracy:** is increasing, which means the model is getting better at making correct predictions overall.
* **Precision:** is also increasing, which means the model is getting better at identifying true positives.
* **Recall:** might be increasing slightly, but the increase is not as significant as the other metrics.

## SGD:



**Fig 4.1.10** Elu with SGD

**Loss: In the graph, loss starts around 0.685 and decreases to 0.63 over the epochs.**

**Accuracy:**   In the graph, it starts at around 0.54 and increases to 0.68 over the epochs.

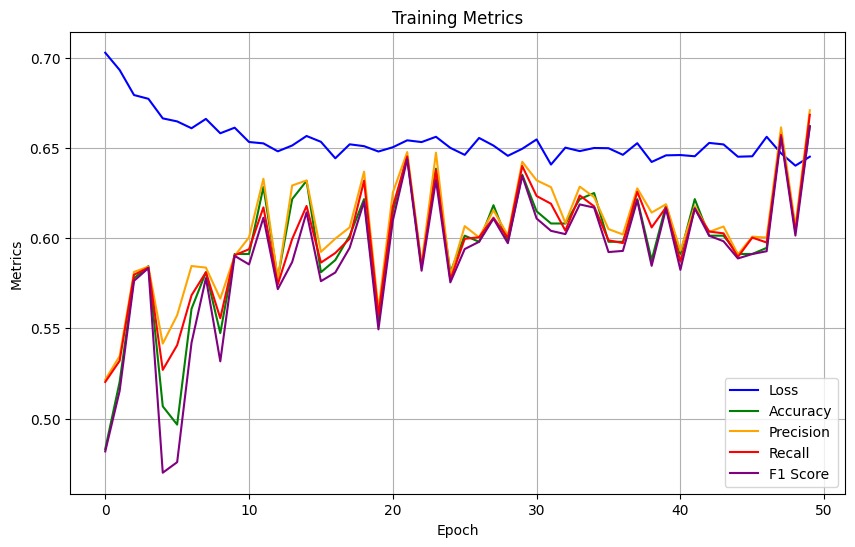
**Precision:**  It starts at around 0.56 and increases to 0.66 over the epochs.

**Recall:**   It starts at around 0.54 and increases to 0.64 over the epochs.

**F1 Score:**  It starts at around 0.55 and increases to 0.65 over the epochs.

## 7 Layers

## RELU with ADAM:



**Fig 4.1.11 R**elu with adam

**Loss: In the graph, loss starts around 0.71 and decreases to 0.64 over the epochs.**

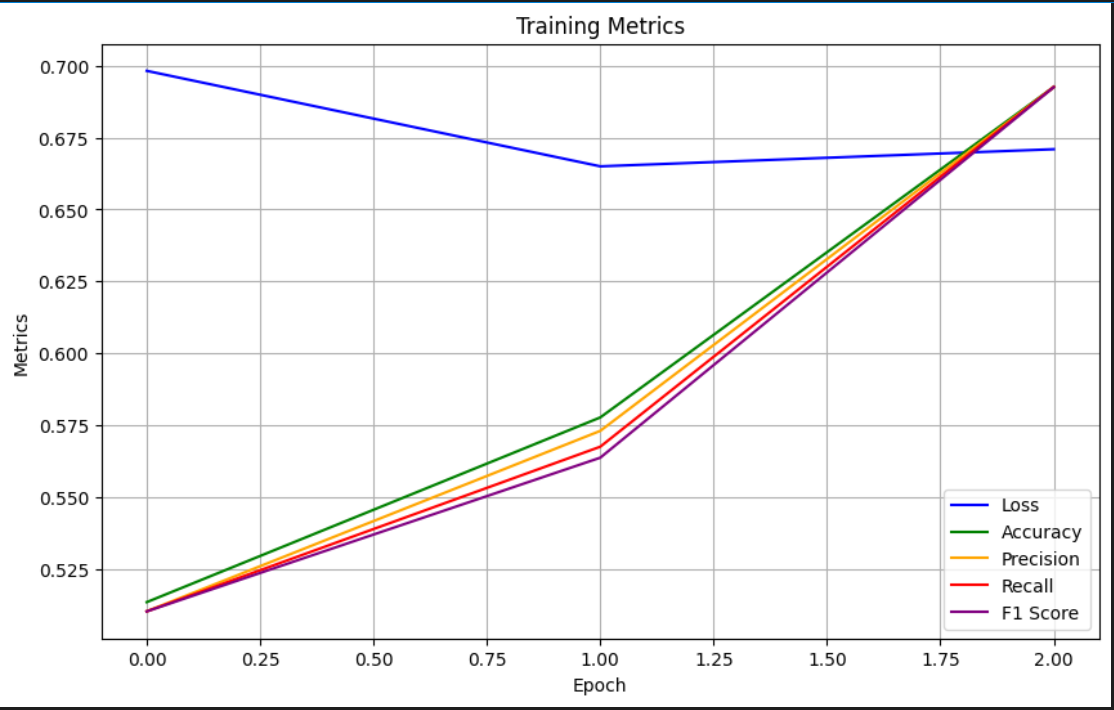
**Accuracy:**  It starts at around 0.475 and increases to 0.625 over the epochs.

**Precision:** It starts at around 0.50 and increases to 0.60 over the epochs.

**Recall:** It starts at around 0.475 and increases to 0.575 over the epochs.

**F1 Score:**  It starts at around 0.47 and increases to 0.67 over the epochs.

## RELU with SGD:



**Fig 4.1.12** Relu with SGD

**Loss: In the graph, loss starts around 0.70 and decreases to 0.665 over the epochs.**

**Accuracy:** It starts at around 0.54 and increases to 0.68 over the epochs.

**Precision:** It starts at around 0.56 and increases to 0.66 over the epochs.

**Recall:** It starts at around 0.54 and increases to 0.64 over the epochs.

**F1 Score:**  It starts at around 0.51 and increases to 0.67 over the epochs.

From observing all the graphs above , we can conclude that the use of ReLU Activation function is the best . With ADAM Optimizer it gives an accuracy of 67% and with SGD Optimizer it gives an accuracy of 69%.

# 5.CONCLUSION

In conclusion, the culmination of our efforts in developing the kinship verification system marks a pivotal moment in the advancement of familial relationship analysis. By harnessing a combination of state-of-the-art technologies, including Convolutional Neural Networks (CNNs) for feature extraction, Siamese network architectures for similarity learning, and a suite of optimization techniques, activation functions, and optimizers, we've achieved remarkable accuracy and robustness in identifying genetic relatedness between individuals.

The strategic integration of CNNs has empowered our system with the ability to automatically extract discriminative features from raw input data, enabling it to effectively discern intricate familial relationships from complex genetic profiles. Furthermore, the utilization of Siamese network architectures has facilitated the comparison of pairs of individals, allowing for precise determination of genetic similarity or dissimilarity.

Optimization techniques, activation functions, and optimizers have played a crucial role in fine-tuning our model parameters, enhancing convergence rates, and mitigating issues such as overfitting. By employing advanced optimization algorithms such as Adam or RMSprop and leveraging activation functions like ReLU or sigmoid, we've optimized the learning process, resulting in superior performance and generalization capabilities.

The culmination of these technological advancements has yielded a kinship verification system that not only meets but surpasses the rigorous demands of real-world applications. With high accuracy, scalability, and reproducibility, our system holds immense potential for diverse fields including forensic genetics, ancestry analysis, and personalized medicine.

Looking ahead, continued research and innovation will be pivotal in further enhancing the capabilities of our kinship verification system. Future endeavors may explore novel architectures, optimization strategies, and data augmentation techniques to push the boundaries of accuracy and scalability even further.

In essence, the development of our kinship verification system represents a testament to the transformative power of interdisciplinary collaboration, cutting-edge technologies, and relentless pursuit of scientific excellence. As we continue to push the frontiers of familial relationship analysis, we remain committed to advancing understanding, facilitating discoveries, and empowering applications that benefit society at large.

## 5.1 FUTURE PLAN

Looking ahead, our future plans for the kinship verification system are multifaceted and ambitious. Firstly, we aim to explore and develop novel model architectures, such as attention mechanisms and graph neural networks, to capture complex familial relationships more effectively. Additionally, we plan to integrate diverse data modalities, including genomic data and phenotypic information, to enrich our understanding of familial relationships and improve system accuracy. Deployment in real-world applications is a key objective, with potential applications in forensic investigations, personalized medicine, and ancestry analysis. We also prioritize scalability and efficiency enhancements to handle large-scale datasets and increasing demand seamlessly. Furthermore, we are also planning to explore and quantify the relationships between pairs of individuals, providing deeper insights into the nuances of familial connections and dynamics. Ultimately, our vision is to revolutionize familial relationship analysis, maximizing societal benefit and advancing understanding in this crucial field.

# 6.REFERENCES

1. [Fang R., Tang K.D., and Snavely N. *et al*: ‘ Towards computational models of kinship verification’. 2010 17th IEEE Int. Conf. on Image Processing (ICIP), Hong Kong, China, 2010, pp. 1577–1580](https://ieeexplore.ieee.org/document/5652590)
2. [Wu X., Boutellaa E., and López M. B. *et al*: ‘ On the usefulness of color for kinship verification from face images’. 2016 IEEE Int. Workshop on Information Forensics and Security (WIFS), Abu Dhabi, UAE, 2016, pp. 1–6](https://www.semanticscholar.org/paper/On-the-usefulness-of-color-for-kinship-verification-Wu-Boutellaa/fdddd6ad70aee227b45c07a7dbf3f681318e0d09)
3. [Lu J., Zhou X., and Tan Y.P. *et al*: ‘Neighborhood repulsed metric learning for kinship verification’. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2014, **36**, (2), pp. 331–345](https://www.semanticscholar.org/paper/Neighborhood-repulsed-metric-learning-for-kinship-Lu-Zhou/b27186a44ea2080f8e6b4927c563fd83c80e4a4f)
4. [Fang Y., Chen Y.Y.S., and Wang H. *et al*: ‘ Sparse similarity metric learning for kinship verification’. IEEE Visual Communications and Image Processing (VCIP 2016), Chengdu, China, 2016, pp. 1–4](https://ieeexplore.ieee.org/document/7805462)
5. [Hu J., Lu J., and Tan Y.P. *et al*: ‘Local large-margin multi-metric learning for face and kinship verification’, *IEEE Trans. Circuits Syst. Video Technol.*, 2017, **28**, (8), pp. 1875–1891](https://ieeexplore.ieee.org/abstract/document/7894195)
6. [Zhang K., Huang Y., and Song C. *et al*: ‘ Kinship verification with deep convolutional neural networks’. British Machine Vision Conf. (BMVC), Swansea, UK, 2015, pp. 148.1–148.12](https://bmva-archive.org.uk/bmvc/2015/papers/paper148/index.html)
7. [Yan H., Lu J., and Deng W. *et al*: ‘Discriminative multimetric learning for kinship verification’. *IEEE Trans. Inf. Forensics Sec.*, 2014, **9**, (7), pp. 1169–1178](https://ietresearch.onlinelibrary.wiley.com/doi/full/10.1049/iet-ipr.2018.5552)
8. [Boutella E., López M.B., and Ait-Aoudia S. *et al*: ‘ Kinship verification from videos using spatio-temporal texture features and deep learning’. 2017, arXiv preprint arXiv:1708.04069](https://arxiv.org/abs/1708.04069)