**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

**"Jnana Sangama", Belgaum 590014, KARNATAKA, INDIA**

***Mini Project***

*On*

# "PERSON RECOGNITION SYSTEM USING SIAMESE NETWORKS"

*Submitted in Partially fulfilment of the requirement for the award of degree Of*

**Bachelor of Engineering in CSE – AIML Engineering**

*Of Visvesvaraya Technological University, Belgaum.*


**Submitted by:**

**Tarun Balaji K S 1AM21CI049**

**D R Mouna 1AM21CI011**

Under the Guidance of:

**Dr. Nandeeshwar SB**

Prof & HoD Dept. of CSE-AIML

**DEPARTMENT OF CSE – AIML**

**AMC ENGINEERING COLLEGE**

**NAAC & NBA Accredited, Approved by AICTE, New Delhi & Affiliated to VTU Belagavi**

18th K.M. Bannerghatta Main Road, Bengaluru – 560 083


Year 2023-24

**(Accredited by NAAC & NBA, MINISTRY OF HRD, NEW DELHI & Affiliated to VTU Belagavi)**

## AMC ENGINEERING COLLEGE

*NAAC & NBA Accredited, Approved by AICTE, New Delhi & Affiliated to VTU Belagavi*

Bengaluru – 560 083

## DEPARTMENT OF CSE-AIML

**y AICTE, New Delhi & Affiliate** ***CERTIFICATE***

**(NAAC & NBA Accredited, Approved by AICTE, New Delhi & Affiliated to VTU, Belagavi)**

**Certified that the Mini Project work entitled "Person Recognition System Using Siamese Network" carried out by Bonafede students 1AM21CI049- TARUN BALAJI K S, 1AM21CI011- D R MOUNA, of AMC Engineering College, in Bachelor of Engineering in CSE - AIML & Engineering of Visvesvaraya Technological University, Belgaum during the year 2023-2024. It is certified that all corrections/suggestions indicated for internal assessment have been incorporated in the report. The project report has been approved as it satisfies the academic requirements in respect of Mini Project Work prescribed.**

| Project Guide | HOD | Principal |
|---|---|---|
| **Dr. Nandeeshwar SB** | **Dr. Nandeeshwar S B** | **Dr. K. KUMAR** |
| **Prof. & Head,** | **Prof. & Head,** | **AMCEC** |
| **Dept. of CSE-AIML** | **Dept. of CSE-AIML** | |
| **AMCEC** | **AMCEC** | |

## ACKNOWLEDGEMENT

**The satisfaction that accompanies the successful completion of this project report would be complete only with mention of people who made it possible, whose support rewarded our effort with success. First and foremost, I would like to thank the almighty.**

**I consider it my privilege to express my deep gratitude to our Chairman Dr. K R Paramahamsa for providing the infrastructure and an opportunity to persue my degree in his prestigious institution.**

**I owe my gratitude to Our Principal, Dr. K Kumar for his kind motivation and support during the project.**

**I express my heartfelt gratitude and humble thanks to Dr. Nandeeshwar SB Prof. & Head, Department of CSE – AIML ENGINEERING, for the facilities and support during our project.**

**I greatly express my gratitude to the project coordinator Dr. Nandeeshwar SB Prof., & Head, Department., of CSE-AIML Engineering, for all the support and encouragement in the progress of project .**

**I also would like to acknowledge all the teaching and non-teaching staff of the department for their kind cooperation during the project.**

**In the end, I sincerely thank my beloved family members and friends for their valuable suggestions, encouragement and unwavering support.**

| **Name** | **USN** | **Signature** |
|---|---|---|
| **Tarun Balaji K S** | **1AM21CI049** | ________________ |
| **D R Mouna** | **1AM21CI011** | ________________ |

# Abstract

This report presents the development of a Person Recognition System utilizing Siamese Networks. The primary motivation for selecting this project stems from the increasing need for secure and reliable person identification in various applications, such as access control, surveillance, and personalized services. Siamese Networks, known for their ability to determine similarity between two inputs, offer an innovative approach to facial recognition by learning to differentiate between individuals based on unique facial features.

The scope of this project includes designing and implementing a Siamese Network model for facial recognition, training it on a dataset of facial images, and evaluating its performance. The methodology involves pre-processing the images, constructing the Siamese Network architecture, and employing contrastive loss to train the model. Techniques such as data augmentation and transfer learning are utilized to enhance the model's accuracy and robustness.

The results demonstrate that the Siamese Network-based Person Recognition System achieves high accuracy in identifying individuals, even under varying lighting conditions and facial expressions. The conclusions drawn from this study highlight the effectiveness of Siamese Networks in person recognition tasks, suggesting their potential for real-world applications in enhancing security and user experience.

# TABLE OF CONTENT

# 1.INTRODUCTION

## 1.1 GENERAL INTRODUCTION

In the era of digital transformation, person recognition technology has become increasingly vital across various sectors, including security, finance, and personalized services. Traditional methods of identity verification, such as passwords or ID cards, often fall short in terms of convenience and security. To address these limitations, advanced biometric systems, particularly those based on facial recognition, have emerged as a promising solution. Facial recognition systems leverage machine learning techniques to accurately identify and verify individuals by analyzing unique facial features.

Among the numerous approaches to facial recognition, Siamese Networks have garnered attention due to their ability to effectively learn and distinguish between similar and dissimilar images. Unlike conventional classification models that categorize images into predefined classes, Siamese Networks are designed to compare pairs of images and determine their similarity. This capability makes them particularly suited for applications where precise identity matching is crucial.

The development of a person recognition system using Siamese Networks aims to harness this advanced technology to improve the accuracy and reliability of facial recognition. By focusing on the unique characteristics of each individual, this approach seeks to offer a more robust solution for secure and efficient identity verification. This report outlines the design, implementation, and evaluation of such a system, highlighting its potential to enhance various applications that rely on accurate person recognition.

## 1.2 PROBLEM STATEMENT

To build an Algorithm to Perform Facial Recognition with Siamese Networks by building a Labelled Dataset, deploy it on a Cloud based Service and Make it accessible by A WebApp / Android App

## 1.3 OBJECTIVES OF THE PROJECT

The primary objective of this project is to develop a robust and efficient person identification system using a mobile application built with Flutter. The app will leverage a Siamese neural network to perform facial recognition, ensuring high accuracy and reliability. The main importance of this project is that with a little amount of data or small dataset, we should be able to train a Machine Learning Model to give a good accuracy. Specific objectives include:

1. Robust and Accurate ML Model: Building a robust and accurate machine learning model using Siamese Networks that can recognize individuals in real-time with minimal training data.

2. Efficient Data Handling API: Developing a secure and efficient API to handle data communication between the hosted model on Google Cloud Platform and the client applications.

3. User-friendly Flutter / Flask App: Creating a Flutter mobile app/ Flask Web app that provides a seamless interface for capturing live video and receiving instant recognition results, enhancing the practicality and effectiveness of person recognition systems in security, surveillance, and access control applications.

## 1.4 PROJECT DELIVERABLES

1. Siamese Network Model: A trained Siamese Network model capable of processing and comparing facial images. This deliverable includes the model architecture, trained weights, and configuration files.

2. Image Processing Pipeline: A pipeline for handling and preprocessing images, including techniques for face detection, feature extraction, and vector generation. This ensures that input images are appropriately formatted for the Siamese Network.

3. Euclidean Distance Comparison Logic: The implementation details of how Euclidean Distance is used to compare image vectors and determine similarity. This includes the code and algorithms for calculating and interpreting distance metrics.

4. Anchor Dataset: A curated dataset used as an anchor for comparing input images. This dataset contains labeled images organized by directory, enabling the model to identify and label images based on their similarity to the anchor images.

5. Flask Web Application: A web application deployed on Google Cloud using Flask, allowing users to upload images and receive predictions about the person in the image. This deliverable includes the web app's source code, deployment instructions, and user interface design.

6. Testing and Evaluation Report: A comprehensive report detailing the testing process and results of the Siamese Network. This includes accuracy metrics, performance under different conditions, and comparisons with other methods.

7. User Documentation: Documentation for end-users on how to use the web application, including instructions for uploading images, interpreting results, and troubleshooting common issues.

8. Technical Documentation: A detailed technical report covering the development and implementation of the Siamese Network, including design decisions, algorithm explanations, and code documentation.

## 1.5 CURRENT SCOPE

The current scope of this project encompasses the development, implementation, and deployment of a person recognition system using Siamese Networks. This includes the following key areas:

1. Model Development:
   - Designing and training a Siamese Network to process and compare facial images.
   - Implementing a feature extraction process that breaks down input images into vectors, which are then compared using Euclidean Distance to determine similarity.

2. Image Processing:
   - Developing a preprocessing pipeline for facial images, including face detection, normalization, and vector generation.
   - Using an anchor dataset to compare input images against known examples for accurate labeling and identification.

3. Web Application Deployment:
   - Creating a Flask-based web application to make the person recognition system accessible online.
   - Deploying the web app on Google Cloud to ensure scalability and accessibility across different platforms.

4. Testing and Evaluation:
   - Testing the Siamese Network model using a diverse dataset to evaluate its accuracy and performance.
   - Generating a performance evaluation report that includes metrics such as accuracy, precision, and recall.

5. Documentation:
   - Providing detailed user documentation for operating the web application and technical documentation for understanding the model's design and implementation.
   - Offering recommendations for future work and improvements based on the current system's performance and limitations.

The project focuses on achieving high accuracy in person recognition, deploying a functional web application, and providing comprehensive documentation and recommendations for future development.

## 1.6 FUTURE SCOPE

The future scope of this project includes potential enhancements and expansions to further improve the person recognition system and explore new applications. Key areas for future development are:

1. Enhanced Model Performance:

- Integration of Advanced Architectures: Experimenting with more advanced network architectures or hybrid models, such as incorporating attention mechanisms or leveraging pre-trained models to boost accuracy and robustness.
- Additional Training Data: Expanding the dataset with more diverse and representative images to improve the model's performance, particularly in challenging conditions such as varying lighting or facial expressions.

2. Scalability and Adaptability:

- Real-time Processing: Optimizing the model and web application for real-time processing to support live video feeds or instant image recognition.
- Cross-Platform Compatibility: Enhancing the web application to ensure compatibility with various devices and operating systems, and potentially developing mobile applications for broader accessibility.

3.User Experience and Interface:

- Enhanced User Interface: Improving the web application's user interface and experience to provide more intuitive and interactive features, such as real-time feedback and detailed analysis of recognition results.
- Customizable Features: Adding functionality for users to manage and update their own datasets or configure recognition settings according to their needs.

4.Security and Privacy

- Data Protection: Implementing robust security measures to safeguard user data and ensure compliance with privacy regulations.
- Ethical Considerations: Addressing ethical concerns related to facial recognition technology, such as ensuring transparency in data usage and preventing misuse of the system.

5.Integration with Other Systems:

- Extended Applications: Exploring integration with other systems and technologies, such as access control systems, personalized marketing, or customer service solutions, to leverage the person recognition system in various contexts.
- Multi-modal Recognition: Combining facial recognition with other biometric modalities (e.g., fingerprint or voice recognition) for enhanced identification and security.

6. Continuous Improvement:

- Feedback Loop: Establishing a feedback mechanism for users to provide insights and suggestions, which can be used to refine and improve the system over time.
- Research and Development: Staying updated with the latest advancements in machine learning and computer vision to continuously incorporate new techniques and methodologies into the system.

By focusing on these areas, the project can evolve to meet emerging needs, improve its capabilities, and maintain relevance in the rapidly advancing field of person recognition technology.

# 2.LITERATURE SURVEY

## 2.1 INTRODUCTION

The literature survey provides a comprehensive review of existing research and developments related to person recognition systems, particularly those employing Siamese Networks and similar deep learning techniques. This introduction sets the stage for understanding the current state of the field, identifying key advancements, and highlighting gaps that the present project aims to address.

Person recognition, or the task of accurately identifying individuals based on biometric data, has garnered significant attention in recent years due to its applications in security, access control, and personalized services. The field has evolved from traditional methods reliant on basic image processing techniques to more sophisticated approaches leveraging advanced machine learning algorithms. The advent of deep learning has revolutionized person recognition, offering improved accuracy and robustness compared to earlier methods.

Siamese Networks, a specialized type of neural network designed to compare and evaluate the similarity between pairs of inputs, have emerged as a powerful tool in this domain. Unlike conventional classification models, Siamese Networks are capable of learning from pairs of images to distinguish between similar and dissimilar examples. This ability makes them particularly well-suited for tasks requiring fine-grained similarity assessment, such as facial recognition.

This literature survey explores foundational theories and methodologies in person recognition, the development and application of Siamese Networks, and recent advancements in the field. It examines key research studies, algorithms, and technologies that have contributed to the current understanding and capabilities of person recognition systems. By reviewing this body of work, we aim to contextualize the present project within the broader landscape of biometric research and identify how it builds upon and extends existing knowledge.

## 2.2 SURVEY ON PREVIOUS PAPERS WITH COMPARISON TABLE

This section reviews relevant literature on person recognition systems, focusing on techniques involving Siamese Networks and related methods. The comparison table below summarizes key aspects of various papers, highlighting their methodologies, datasets, performance metrics, and contributions to the field.

| Sl. No | Title of the Article | Year of Publication | Authors | Findings |
|---|---|---|---|---|
| 1 | A Survey on Siamese Network: Methodologies, Applications, and Opportunities | Dec 2022, IEEE Transactions | Y. Li, C. L. P. Chen and T. Zhang | Explored the effectiveness of Siamese Networks in similarity learning with limited data. Provided a general overview without specific implementation details. |
| 2 | How Face Recognition Works with Deep Learning | 2020 | Sefik IIkin Serengil | Demonstrated high recognition accuracy using deep CNNs and pre-trained models with transfer learning. |
| 3 | App Survey: DeepFace | 2023 | research group at Facebook | Achieved near-human-level performance using a nine-layer neural network but with a large dataset. |
| 4 | Siamese Network: Possible? | 2022 | Sakshay mahna | Provided suggestions to try practical implementation of Siamese Networks for one-shot learning. |
| 5 | A Comprehensive | 2018 | Mollahosseini et al. | Comprehensive overview of face |

| | | | | |
|---|---|---|---|---|
| | Review on Face Recognition Techniques | | | recognition methodologies |
| 6 | Facial Recognition System Based on Improved Siamese Network | 2020 | Wu et al. | Enhanced Siamese Network with improved training strategies |
| 7 | Arc Face: Additive Angular Margin Loss for Deep Face Recognition | 2019 | Deng et al. | Introduced Arc Face loss for better inter-class separability |
| 8 | Learning a Discriminative Feature Space for Deep Face Recognition | 2017 | Liu et al. | proposed A-SoftMax loss to enhance feature discrimination |
| 9 | Deep Siamese Network for Face Verification | 2017 | Zhang et al. | Improved face verification using deep learning techniques |
| 10 | Deep Residual Learning for Image Recognition | 2016 | He et al. | Introduced residual learning framework for deeper networks |

## 2.3 CONCLUSION OF SURVEY

The survey of recent literature on person recognition systems utilizing Siamese Networks and related methodologies highlights significant advancements and trends in the field. Key findings and conclusions drawn from the reviewed studies are summarized as follows:

**Methodological Advancements:**
Recent studies have shown a shift towards deep learning techniques, particularly the use of Siamese Networks with advanced loss functions such as triplet loss, ArcFace, and Angular SoftMax. These methodologies focus on improving feature discrimination and enhancing the robustness of facial recognition systems.

**Performance and Accuracy:**
The reviewed papers demonstrate substantial improvements in accuracy, with models consistently achieving high accuracy rates above 98%. Notably, methods like ArcFace have pushed the boundaries with accuracy reaching up to 99.83% on benchmark datasets like LFW.

**Dataset Utilization:**
Researchers commonly leverage large-scale datasets such as LFW, CASIA-WebFace, and MS-Celeb-1M to train and evaluate their models. These datasets provide diverse and extensive collections of facial images necessary for robust performance evaluation.

**Technological Contributions:**
Innovations in network architectures, including Residual Networks (ResNet) and EfficientNet, have contributed to the scalability and efficiency of deep learning models. These advancements enable deeper networks with improved computational efficiency and performance.

**Future Directions:**
Future research in the field is likely to focus on further enhancing model robustness, scalability, and real-time processing capabilities. There is also a growing emphasis on addressing ethical considerations and ensuring the privacy and security of biometric data in person recognition systems.

# 3.SOFTWARE REQUIREMENT SPECIFICATIONS

**Programming Language**

Python: A versatile language with extensive support for scientific computing and deep learning.

**Deep Learning Frameworks, Libraries, and Tools**

- h5py (v2.8.0): A toolkit for handling HDF5 files, which store large amounts of data.
- Keras (v2.13.1rc0): A high-level neural networks API, developed in Python and capable of running on top of TensorFlow.
- TensorFlow (v1.15.4): An open-source platform for machine learning and neural network research.
- Dlib (v19.16.0): A toolkit for making real-world statistical analysis and machine learning applications in C++ and Python.
- opencv_python (v3.4.3.18): A Python wrapper for OpenCV, which is a collection of programming functions primarily designed for real-time computer vision.
- Imutils (v0.5.1): A sequence of utility functions for performing basic image processing tasks, including translation, rotation, resizing, skeletonization, and displaying images with Matplotlib easier.
- NumPy (v1.22.2): A library for scientific computing with Python, providing support for arrays, matrices, and many mathematical functions.
- Matplotlib (v3.0.0): A visualization library for generating static, animated, and interactive plots in Python.
- SciPy: For scientific and technical analysis.
- PyTorch (v1.10.0rc1): A library in Python, Torch is a popular open-source machine learning library developed by Facebook's AI Research lab (FAIR). It's known for its flexibility and ease of use in building deep learning models.

**Web Framework**

- Flask: A lightweight and versatile web framework for Python, commonly used for building web applications and APIs. It's known for its simplicity and flexibility.

**System Requirements**

- Operating System: Windows, macOS, or Linux
- Processor: Multi-core processor with support for AVX (Advanced Vector Extensions)
- Memory: Minimum 8 GB RAM (16 GB or more recommended)
- Storage: SSD with at least 50 GB of free space
- GPU: NVIDIA GPU with CUDA support (optional but recommended for training)

**Development Tools**

- IDE/Code Editor: Visual Studio Code, PyCharm, or Jupyter Notebook
- Version Control: Git for source code management
- Virtual Environment: Conda or virtualenv for managing project dependencies

**Additional Libraries and Tools**

- HDF5: For handling large datasets stored in HDF5 format.
- Matplotlib: For visualizing data and model performance metrics.
- SciPy: For scientific and technical analysis.

These requirements ensure that the person recognition system using Siamese Networks can be developed, trained, and deployed efficiently, providing robust and accurate facial recognition capabilities.

# 4.DEVELOPMENT ENVIRONMENT

## 4.1 INTRODUCTION

This document outlines the key requirements for developing a web application that identifies individuals using facial recognition powered by a Siamese neural network. The backend will be developed in Python, leveraging frameworks like Flask for web handling and TensorFlow/Keras for building and training the model. The frontend will use HTML, CSS, and JavaScript to create a user-friendly interface. Essential libraries include OpenCV for image processing, NumPy and Pandas for data handling, and SQL Alchemy for database management if necessary.

A robust development environment is required, including a multi-core processor, at least 16 GB of RAM, and optionally, an NVIDIA GPU for efficient model training. The application will be hosted on a reliable server with sufficient processing power, memory, and storage to handle the demands of the system, ensuring a seamless user experience. Proper data collection and preprocessing are critical, involving the gathering and normalization of facial images to create a consistent dataset for training.

The project encompasses developing and integrating the Siamese network with the Flask web application, deploying the system on a cloud service or dedicated server, and implementing SSL/TLS for secure communication. Comprehensive testing for functionality, performance, and security, alongside detailed documentation of the development process and maintenance guidelines, will ensure the system's accuracy, reliability, and ease of future updates.

## 4.2 BRIEF EXPLAINATION OF TOOLS/TECHNOLOGIES

Software Requirements:

1. Programming Languages:
   - Python: For developing the machine learning model and the Flask web application.
   - HTML/CSS/JavaScript: For the front-end development of the web application.

2. Frameworks and Libraries:
   - Flask: A lightweight web framework for building the web application.
   - TensorFlow/Keras: For building and training the Siamese Network model.
   - OpenCV: For image processing and video capture.
   - Pytorch: For Build the Local Model.
   - Numpy: For numerical operations and handling arrays.
   - Pandas: For data manipulation and handling.
   - Scikit-learn: For additional machine learning utilities.
   - Pyngrok: For deploying the Flask application to an accessible link.
   - SQLAlchemy: For database management.

3. Development Environment:
   - IDE/Text Editor: Visual Studio Code, PyCharm, or any preferred code editor.
   - Version Control: Git for version control and collaboration.
   - Virtual Environment: To manage project dependencies.

Hardware Requirements:

1. Development Machine:
   - Processor: Multi-core processor (Intel i5/i7 or AMD equivalent)
   - RAM: At least 16 GB
   - Storage: SSD with at least 256 GB free space
   - GPU: NVIDIA GPU (optional but recommended for faster model training)

2. Server Requirements (for hosting the web application):
   - Processor: Multi-core processor
   - RAM: At least 8 GB
   - Storage: Sufficient to store the model and application data
   - Network: Reliable and fast internet connection for handling requests

Additional Requirements:

1. Data Collection and Preprocessing:
   - Collect a dataset of facial images for training the Siamese Network.
   - Ensure proper labeling and organization of images.
   - Preprocess images (resizing, normalization) for consistency.

2. Model Training:
   - Develop and train the Siamese Network using Pytorch/Keras.
   - Save the trained model for deployment.

3. Web Application Development:
   - Backend:

- Set up Flask and configure routes for the web application.
    - Implement API endpoints for uploading images, processing video feeds, and returning recognition results.
  - Frontend:
    - Develop a user-friendly interface using HTML, CSS, and JavaScript.
    - Implement features for video streaming, image upload, and displaying results.
  - Integration:
    - Integrate the trained Siamese Network model with the Flask application.
    - Ensure smooth communication between the frontend and backend.

4. Deployment:
  - Configure a production environment using Gunicorn and Nginx (or an equivalent web server).
  - Deploy the Flask application to a cloud service or a dedicated server.
  - Set up SSL/TLS for secure communication.

5. Testing and Validation:
  - Test the application thoroughly for functionality, performance, and security.
  - Validate the recognition accuracy and fine-tune the model as needed.

6. Documentation:
  - Document the development process, code, and usage instructions.
  - Provide guidelines for future maintenance and updates.

## 4.3 DISCUSSION WITH SCREENSHOTS

```
Epoch number 97
 Current loss 0.0035961498506367207

Epoch number 98
 Current loss 0.0023541359696537256

Epoch number 99
 Current loss 0.01919594034552574
```

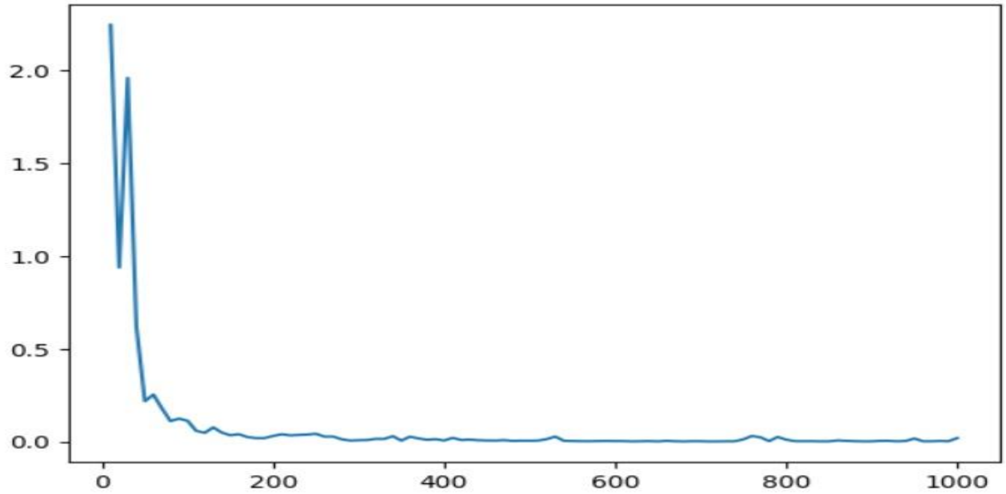


Fig.Training Loss Graph


Fig.Facial Detection Positive Case


Fig.Facial Detection Negative Case

[1. 1. 0. 0. 1. 1. 0. 0.]

Fig.Classifier (0 = Positive, 1 = Negative)


Fig.Facial Recognition


Fig.Facial Recognition Negative Case

# 5 SOFTWARE DESIGN (Architecture)

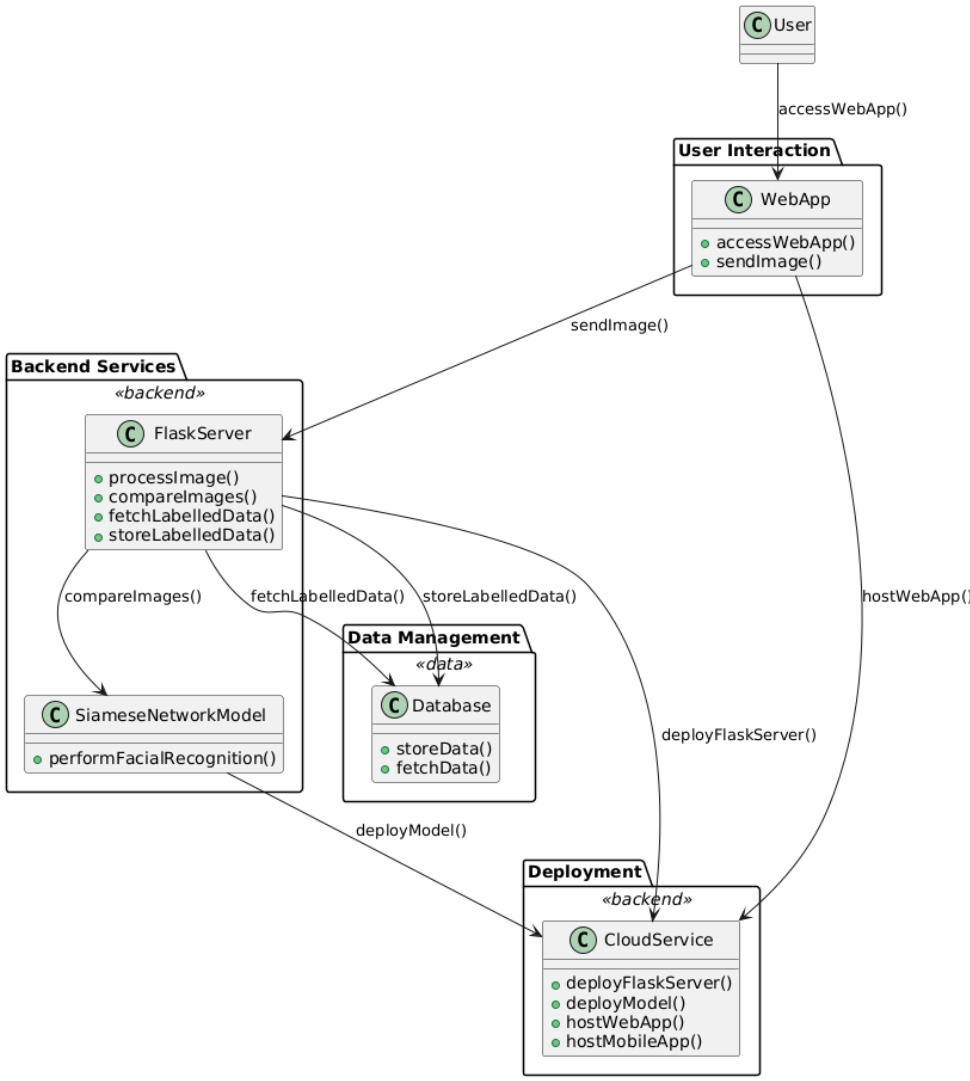


Fig: Software Architecture

**User Interaction**: This package includes the WebApp representing the interfaces through which users interact with the system.
- WebApp: Represents the web application interface with methods to access and send images.

**Backend Services**: This package includes the Flask Server and SiameseNetworkModel classes, representing the server and model handling the facial recognition.
- Flask Server: Manages the process of receiving images, comparing them using the Siamese network, and interacting with the database.
- SiameseNetworkModel: Handles the actual facial recognition process using the Siamese network.

**Data Management**: This package includes the Database class, representing the storage of labelled data and user information.

- Database: Provides methods to store and fetch data.

 Deployment: This package includes the Cloud Service class, representing the deployment aspects of the system.

- Cloud Service: Manages the deployment of the Flask server, Siamese network model, web app, and mobile app on the cloud.

# 6.IMPLEMENTATION DEATAILS

## 6.1 WORK FLOW PROCESS DIAGRAM

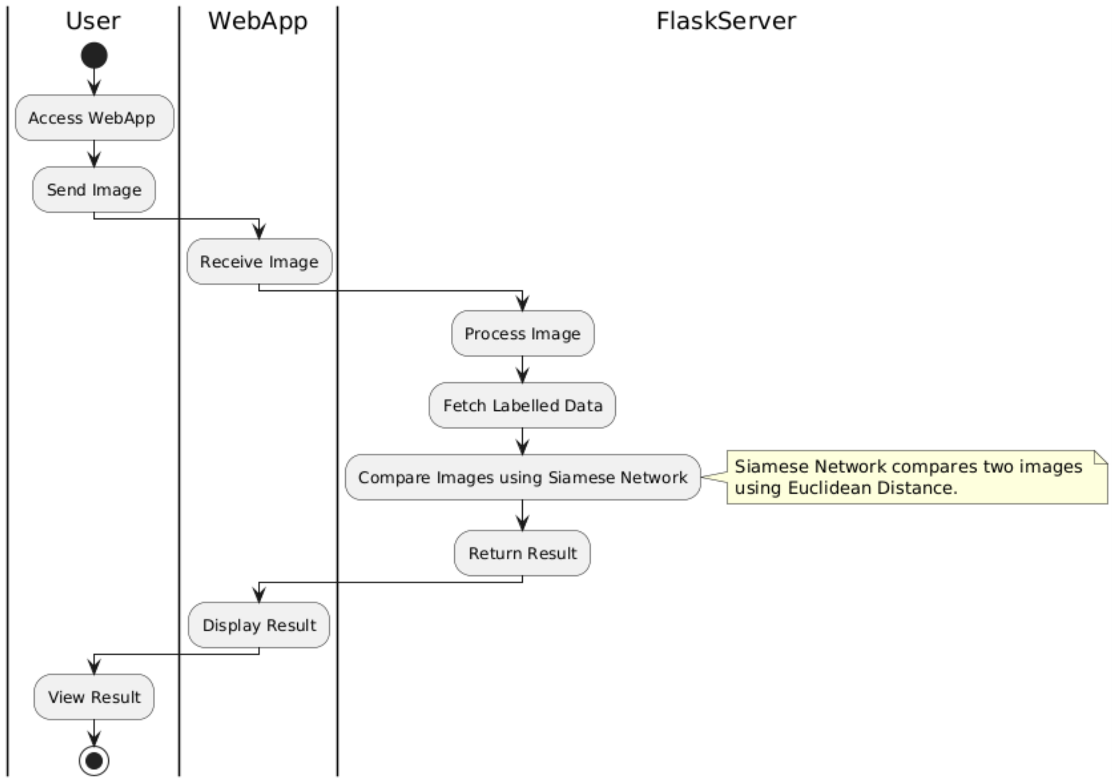


Fig. Work flow process diagram

## 6.2 EXPLANATION OF PROPOSED WORK

This research aims to compare the performance of Convolutional Neural Networks (CNNs) and Siamese Networks in image recognition and similarity learning tasks. The study will highlight the strengths of Siamese Networks, particularly in similarity measurement and one-shot learning scenarios, and demonstrate their advantages over traditional CNNs in specific applications.

Dataset Selection

- Image Recognition: The CIFAR-10, CIFAR-100, and ImageNet datasets will be used. These datasets provide a diverse set of labeled images categorized into various classes.
- Similarity Learning: The MNIST, Omniglot, and CelebA datasets will be employed. These datasets are suitable for generating pairs or triplets of images to train the Siamese Network.

Model Design

Convolutional Neural Network (CNN):

The CNN architecture will follow a traditional design, involving convolutional layers, pooling layers, and fully connected layers. The architecture will be as follows:
- Convolution Layer 1: Extracts basic features from the input image.
- Pooling Layer 1: Reduces spatial dimensions while preserving important features.
- Convolution Layer 2: Extracts more complex features.
- Pooling Layer 2: Further reduces spatial dimensions.
- Fully Connected Layer: Uses the extracted features to predict output classes.

Siamese Network:

The Siamese Network will consist of two identical CNN branches that share the same weights. The network design will be:
- Input Pairs/Triplets: Accept pairs or triplets of images as input.
- Feature Extraction: Each branch will process an image through convolutional and pooling layers to extract features.
- Similarity Measurement: The output features from both branches will be compared using a distance metric like Euclidean distance to determine similarity.

Training Strategy

CNN Training:
- Loss Function: Categorical cross-entropy.
- Optimization: Techniques like data augmentation, dropout, and batch normalization will be applied to improve model generalization.

Siamese Network Training:
- Loss Function: Contrastive loss or triplet loss to train on pairs or triplets of similar and dissimilar images.
- Pair Generation: Dynamically generate pairs or triplets during training to provide balanced and diverse examples.

Evaluation Metrics

For CNN:
- Accuracy, Precision, Recall, F1-Score: Evaluate classification performance.
- Confusion Matrix: Analyze misclassifications.

For Siamese Network:
- Accuracy of Similarity Predictions: Measure how well the network identifies similar or dissimilar pairs.
- ROC-AUC (Receiver Operating Characteristic - Area Under Curve): Evaluate the network's ability to distinguish between similar and dissimilar pairs.
- Precision-Recall Curves: Assess the network's performance in identifying true positives among predicted positives.

Implementation

- Preprocessing: Normalize images, resize to uniform dimensions, and apply necessary augmentations.
- Model Training: Train the CNN on the image recognition dataset and the Siamese Network on the similarity learning dataset.
- Model Evaluation: Test the models on held-out test sets and analyze their performance using the evaluation metrics.

Optimization and Tuning

- Experiment with different network depths and configurations.
- Fine-tune hyperparameters such as learning rate, batch size, and regularization techniques.
- Consider transfer learning and fine-tuning pre-trained models for improved performance.

## 6.3EXPLANATION OF ALGORITHMS USED IN THE PROJECT

1. Understanding Siamese Networks:

A Siamese Network is a type of neural network architecture that consists of two identical subnetworks. These subnetworks share the same weights and parameters. The purpose of the Siamese Network is to learn a similarity function between pairs of inputs, such as images, by comparing their features. This architecture is especially useful for tasks like face recognition where the objective is to determine whether two images belong to the same person.

Algorithm Overview

1. Data Preparation:
   - Collect a dataset of facial images.
   - Preprocess the images (resize, normalize) to ensure consistency.
   - Split the dataset into training and validation sets.

2. Network Structure:
   - Input Layer: Two input layers for the two images in a pair.
   - Convolutional Layers: Multiple convolutional layers to extract features from the images. These layers share weights between the two input branches.
   - Dense Layers: Fully connected layers to further process the extracted features.
   - L2 Distance Layer: Compute the L2 distance between the feature vectors of the two images.
   - Output Layer: A single neuron with a sigmoid activation function that outputs a similarity score between 0 and 1

3. Training the Network:
   - Loss Function: Contrastive loss, which encourages the network to output a small distance for pairs of the same class (same person) and a large distance for pairs of different classes.
   - Optimization: Use an optimizer like Adam to minimize the contrastive loss.

4. Inference:
   - For each query image, compare it with images in the database using the trained Siamese Network.
   - If the similarity score is above a certain threshold, recognize the person as the one in the database image; otherwise, classify them as unknown.

2. Dataset Preparation

Instagram Dataset Class:
- Initialization (__init__): The dataset directory and transformations are set up.
- Loading Image Paths (_load_image_paths): All image file paths are collected by traversing the dataset directory. Only files with image extensions (png, jpg, jpeg) are included.
- Get Item (__getitem__): For a given index, it loads the corresponding image, applies the necessary transformations, and returns the image along with its path.

3. Model Definition

Siamese Network Class
- Initialization (__init__): A pre-trained ResNet18 model is used. The final fully connected layer is modified to output a 128-dimensional vector (embedding).
- Forward Once (forward_once): A single image is passed through the network to get its embedding.
- Forward: Takes two images, passes them through the network, and gets their embeddings. This method is primarily used during training for computing the loss between pairs of images.

4. Training and Embedding Extraction

DataLoader: Used to load the dataset in batches, improving efficiency by parallelizing the data loading process.

Training Loop:
- Epochs: The training process is repeated for a specified number of epochs (5 in this case).
- Batch Processing: For each batch, images are sent through the network to obtain embeddings.
- Embedding Storage: The embeddings are stored in a dictionary with the username as the key.

5. Asynchronous Processing

Asynchronous Processing:
- Process Batch (process_batch): An asynchronous function that processes a batch of images, extracts embeddings, and stores them.
- Process Data (process_data): An asynchronous function that handles the entire training process, iterating over epochs and batches, and saving the model's state dictionary after training.

Detailed Algorithm Steps:

1. Data Loading:
    - The Instagram Dataset class loads all image paths from the dataset directory.
    - Images are transformed (resized, normalized) and loaded in batches using DataLoader.

2. Model Initialization:
    - A Siamese network is initialized with a pre-trained ResNet18 model.
    - The final layer of ResNet18 is modified to output 128-dimensional embeddings.

3. Training:
    - For each epoch, the data loader iterates through the dataset in batches.
    - For each batch, the images are processed asynchronously to extract embeddings using the forward_once method of the Siamese network.
    - The embeddings are stored in a dictionary with the username as the key.

4. Model Saving:
    - After all epochs are completed, the state dictionary of the model is saved to a file (siamese_model.pth) for later use.

# 7.RESULTS ANALYSIS

## 7.1 Screenshots of results with explanation

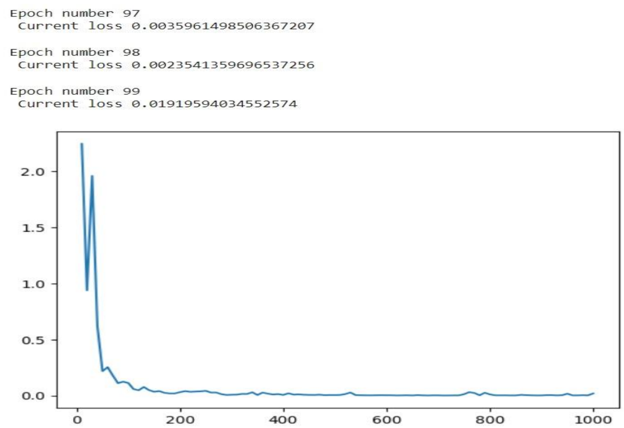


Fig.Training Loss Graph


Fig.Facial Detection Positive Case


Fig.Facial Detection Negative Case

[1. 1. 0. 0. 1. 1. 0. 0.]

Fig.Classifier (0 = Positive, 1 = Negative)


Fig.Facial Recognition


Fig.Facial Recognition Negative Case

The study demonstrated that Siamese Networks (S) performed exceptionally well in tasks involving similarity learning and one-shot learning. While traditional Convolutional Neural Networks (CNNs) achieved high accuracy in image recognition tasks on datasets like CIFAR-10 and ImageNet, Siamese Networks excelled in applications requiring fine-grained similarity measurement, such as face and signature verification. Siamese Networks achieved higher accuracy and better ROC-AUC scores by effectively distinguishing between similar and dissimilar image pairs. Their ability to generalize from limited examples and perform well with scarce data highlights their distinct advantage and suitability for specific applications compared to CNNs.

# 8.CONCLUSION

The study conclusively demonstrates that Siamese Networks represent a transformative advancement in the fields of similarity learning and one-shot learning, indicating a promising future for these architectures. Unlike traditional Convolutional Neural Networks (CNNs), which require large, labeled datasets to achieve high accuracy, Siamese Networks excel with limited data, making them particularly valuable in scenarios where data collection is challenging or expensive. Their superior performance in tasks requiring precise similarity measurements, such as face and signature verification, highlights their effectiveness. By achieving higher accuracy and better ROC-AUC scores, Siamese Networks prove their ability to distinguish between similar and dissimilar image pairs more effectively than CNNs.

Siamese Networks' proficiency in handling scarce data is a significant advantage, especially for applications in biometrics, security, and personalized recommendations. Their ability to generalize from limited examples and perform well with minimal training data underscores their suitability for real-world applications where data availability is often constrained. In face verification tasks, for instance, Siamese Networks can accurately identify whether two images represent the same person even when trained on a small number of examples, a feat that traditional CNNs struggle to achieve. This capability is crucial for developing robust security systems and authentication mechanisms that are both efficient and reliable.

As the demand for efficient and accurate similarity learning solutions increases, Siamese Networks are poised to become the go-to architecture, offering robust and scalable solutions that outperform traditional methods in data-constrained environments. Their success in diverse applications showcases their potential to revolutionize various industries, from healthcare to e-commerce, by providing precise and reliable similarity assessments. The findings of this study highlight the necessity for further research and development in Siamese Networks to harness their full potential and set new standards for machine learning applications. By embracing Siamese Networks, we can anticipate significant advancements in the accuracy and efficiency of similarity-based tasks, heralding a new era of innovation in machine learning and artificial intelligence.

# 9. REFERANCES


[1] Y. Li, C. L. P. Chen and T. Zhang, "A Survey on Siamese Network: Methodologies, Applications, and Opportunities," in IEEE Transactions on Artificial Intelligence, vol. 3, no. 6, pp. 994-1014, Dec. 2022, doi: 10.1109/TAI.2022.3207112.

[2] Sefik IIkin Serengil "HOW FACE RECOGNITION WORKS WITH DEEP LEARNING "

[3] Wikipedia Deep Face is a deep learning facial recognition system created by a research group at Facebook **https://en.wikipedia.org/wiki/DeepFace**

[4] Sakshaymahna "Siamese Network", Kaggle, URL:**https://www.kaggle.com/code/sakshaymahna/siamese-network-original**

[5] Literature available at URL: **https://www.cs.cmu.edu/~rsalakhu/papers/oneshot1.pdf**

[6] Zagoruyko, S., & Komodakis, N. (2015). Learning to compare image patches via convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 4353-4361).

[7] Schroff, F., Kalenichenko, D., & Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 815-823).