Project Report

on

Gait Recognition

A Report of Major Project

B. Tech and M. Tech

in

Mathematics and Data Science

Submitted by

Nayan Awasthi

Scholar No. 214104009

Under the guidance of

Dr. Amit Bhagat

Department of Mathematics, Bioinformatics and Computer Applications

Maulana Azad

National Institute of Technology Bhopal - 462003 (India)

MAULANA AZAD NATIONAL INSTITUTE OF TECHNOLOGY, BHOPAL (M.P)-462003

CERTIFICATE

This is to certify that Nayan Awasthi (Sch no- 214104009), a student of Dual Degree (B.Tech+M.Tech) of batch 2021 -2026 has completed the project titled "Gait Recognition" being submitted in the partial fulfilment of the requirement of the completion of Dual Degree in Mathematics and Data Science to Maulana Azad National Institute of technology Bhopal under my supervision.

Date: 21th April 2024

Place: Bhopal

Dr. Amit Bhagat
Assistant Professor
(Supervisor)

INDEX

Abstract

This project focuses on human identification through gait recognition using deep learning techniques. A Convolutional Neural Network (CNN) model is trained on the CASIA-B dataset, which contains image sequences of 124 individuals walking. The model processes grayscale images to extract gait features, enabling accurate identification of individuals based on their walking patterns. Data augmentation and normalization techniques are employed to improve model generalization. The system achieves promising results in identifying people, demonstrating the potential for gait-based biometric identification in security and surveillance applications, offering a non-intrusive and effective method for human recognition. This project aims to develop a deep learning-based approach for human identification through gait recognition using the CASIA-B dataset. The dataset comprises walking sequences of 124 individuals captured from different views, making it suitable for analyzing gait patterns. We utilize a Convolutional Neural Network (CNN) architecture for feature extraction and classification, where grayscale images of individuals walking are fed into the model. Key techniques such as data augmentation, normalization, and efficient batch processing are applied to enhance the model's performance and generalization. Our results demonstrate that the proposed model can successfully identify individuals based on their walking patterns, achieving strong accuracy. This project highlights the potential of gait recognition as a non-intrusive biometric identification system, with applications in surveillance, security, and other areas where human identification is critical.

Acknowledgement

I take immense pleasure in expressing my deep sense of gratitude to **Dr. Amit Bhagat** and **Dr. Vinod Maliya** for their constant support and encouragement throughout the course of this major project. Their guidance has been invaluable in helping me successfully complete this work.

I would also like to extend my sincere thanks to all the faculty members of the **Mathematics and Data Science (MDS)** department, who have directly or indirectly motivated and encouraged me during this journey.

Finally, I am grateful to my friends for their unwavering moral support and companionship, which played a crucial role in the completion of this project.

INTRODUCTION

1.Relevance of the Project:

Gait recognition is an emerging field with significant applications in security, surveillance, and biometric authentication. Unlike other biometric methods like facial recognition or fingerprinting, gait recognition can identify individuals from a distance without the need for close contact or high-resolution imaging. This makes it highly relevant for real-world applications where unobtrusive identification is crucial. By leveraging deep learning techniques, this project advances the ability to recognize human presence and identify individuals based on their walking patterns, enhancing automated security systems. Gait recognition is an emerging field with significant applications in security, surveillance, and biometric authentication. Unlike other biometric methods like facial recognition or fingerprinting, gait recognition can identify individuals from a distance without the need for close contact or high-resolution imaging. This makes it highly relevant for real-world applications where unobtrusive identification is crucial. In environments such as airports, public spaces, and restricted areas, gait recognition can enhance security by identifying individuals in motion, even when they are partially obscured or under low-light conditions. By leveraging deep learning techniques, this project advances the ability to recognize human presence and identify individuals based on their walking patterns, enhancing automated security systems.

2. Problem Statement:

Identifying individuals from a distance using biometric data, specifically their walking patterns, remains a challenge due to the complexity of human gait variations across different scenarios. The problem is twofold: first, detecting if an image contains a human; second, recognizing the identity of the individual based on their gait. Existing methods often struggle to capture gait features effectively across varying conditions, making it necessary to explore advanced deep learning models to enhance accuracy in gait-based human recognition systems.

3. Objective of the Project:

The primary objective of this project is to develop and implement a robust system for gait recognition, addressing two specific tasks. First, to detect human presence in images using deep learning models like CNNs. Second, to recognize individual identities based on their gait patterns using the CASIA-A dataset, leveraging a variety of models including CNN, ResNet, VGG16, and LSTM. The project aims to identify the most effective model for accurate person identification through gait, with a focus on enhancing LSTM performance for sequential data analysis.

4. Scope of the Project

The project covers two key areas of gait recognition: human detection and individual identification based on walking patterns. Using advanced deep learning techniques, the project explores a wide range of models, such as CNN, ResNet, VGG16, and LSTM, to tackle these tasks. The scope extends to testing and evaluating these models on the CASIA-A dataset for identity recognition, with potential applications in security, surveillance, and biometric systems. Future work could involve expanding to larger datasets and integrating real-time systems for broader use cases. The project covers two key areas of gait recognition: human detection and individual identification based on walking patterns. Using advanced deep learning techniques, the project explores a wide range of models, such as CNN, ResNet, VGG16, and LSTM, to tackle these tasks. The scope extends to testing and evaluating these models on the CASIA-A dataset for identity recognition, with potential applications in security, surveillance, and biometric systems. In addition to improving security systems, the project's scope includes future work on expanding the dataset to cover more diverse scenarios, such as varying lighting, clothing, and movement speeds. It also offers the potential for integrating real-time gait recognition in dynamic environments like smart cities, enhancing monitoring and safety measures on a larger scale.

Process

The project aims to predict laptop prices based on various features using regression models. The problem statement is that if any user wants to buy a laptop, then our application should be compatible to provide a tentative price of laptop according to the user configurations.

The process of the project involves 6 major steps:

1. Data Collection:

The **CASIA-A** dataset used in this project contains images of individuals categorized by IDs, with each individual having 21 subfolders representing various walking conditions. The images in this dataset are captured at different angles, specifically at 0°, 45°, and 90°, to account for variations in the walking posture. These images are used for training and evaluating the gait recognition model, with the aim to handle diverse walking conditions and angles. The CASIA-A dataset can be accessed at the CASIA Gait Database page.

2. Data Preprocessing:

Preprocessing is a crucial step for preparing the data for model training. The CASIA-B dataset consists of images that need to be resized, normalized, and formatted appropriately for use in a convolutional neural network (CNN). The following steps were carried out:

- **Resizing Images:** The images were resized to a consistent shape to ensure uniform input to the CNN model.
- **Normalization:** The pixel values of the images were normalized to a scale of 0-1 by dividing by 255.
- **Label Encoding:** The dataset is divided into 21 classes, with each individual being represented by a unique label (from fyc to zyf).

3. Exploratory Data Analysis (EDA)

In this step, the dataset is explored to understand its structure and gain insights into its distribution. Key aspects like the number of images per individual, the variety of walking conditions, and the class imbalance (if any) are examined. Visualizations such as histograms, box plots, and sample images are used to detect patterns, trends, and any data quality issues that need attention before model training.

4. Model Selection

After data preprocessing, the choice of machine learning models is crucial. For Task 1 (human detection), a **Convolutional Neural Network (CNN)** is chosen due to its effectiveness in image classification. For Task 2 (identity recognition), more advanced models are selected, including **ResNet**, **VGG16**, and **Long Short-Term Memory (LSTM)** networks. Each model's strengths, such as CNN's ability to detect spatial features or LSTM's capability to capture temporal sequences, are leveraged to achieve optimal results.

A. CNN MODEL

Model: "sequential_6"

Layer (type)	Output Shape	Param #
conv2d_5 (Conv2D)	(None, 222, 222, 32)	896
max_pooling2d_5 (MaxPooling2D)	(None, 111, 111, 32)	0
conv2d_6 (Conv2D)	(None, 109, 109, 64)	18,496
max_pooling2d_6 (MaxPooling2D)	(None, 54, 54, 64)	0
flatten_1 (Flatten)	(None, 186624)	0
dense_14 (Dense)	(None, 128)	23,888,000
dropout_7 (Dropout)	(None, 128)	0
dense_15 (Dense)	(None, 20)	2,580

Total params: 23,909,972 (91.21 MB)
Trainable params: 23,909,972 (91.21 MB)
Non-trainable params: 0 (0.00 B)

B. VGG16

Model: "functional_1"

Layer (type)	Output Shape	Param #
input_layer_5 (InputLayer)	(None, 224, 224, 3)	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1,792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36,928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73,856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147,584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295,168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590,080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590,080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1,180,160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2,359,808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2,359,808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2,359,808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2,359,808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
global_average_pooling2d_4 (GlobalAveragePooling2D)	(None, 512)	0
dense_10 (Dense)	(None, 128)	65,664
dropout_5 (Dropout)	(None, 128)	0
dense_11 (Dense)	(None, 20)	2,580

Total params: 14,782,932 (56.39 MB)
Trainable params: 68,244 (266.58 KB)
Non-trainable params: 14,714,688 (56.13 MB)

C. ResNET

```
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Dense, Dropout, GlobalAveragePooling2D
from tensorflow.keras.optimizers import Adam

# Input shape explicitly defined
input_shape = (224, 224, 3)

# Load ResNet50 base model with pre-trained weights
base_model = ResNet50(weights='imagenet', include_top=False, input_shape=input_shape)

# Freeze the base model layers
for layer in base_model.layers:
    layer.trainable = False

# Add custom layers on top of ResNet50
x = base_model.output
x = GlobalAveragePooling2D()(x)
x = Dense(128, activation='relu')(x)
x = Dropout(0.5)(x)
output = Dense(len(train_data.class_indices), activation='softmax')(x)

# Define the full model
model = Model(inputs=base_model.input, outputs=output)

# Compile the model
model.compile(
    optimizer=Adam(learning_rate=0.001),
    loss='categorical_crossentropy',
    metrics=['accuracy']
)
```

D. LSTM

Layer (type)	Output Shape	Param #
conv2d_6 (Conv2D)	(None, 222, 222, 32)	896
max_pooling2d_6 (MaxPooling2D)	(None, 111, 111, 32)	0
conv2d_7 (Conv2D)	(None, 109, 109, 64)	18,496
max_pooling2d_7 (MaxPooling2D)	(None, 54, 54, 64)	0
conv2d_8 (Conv2D)	(None, 52, 52, 128)	73,856
max_pooling2d_8 (MaxPooling2D)	(None, 26, 26, 128)	0
flatten_2 (Flatten)	(None, 86528)	0
dense_12 (Dense)	(None, 128)	11,075,712
dropout_6 (Dropout)	(None, 128)	0
reshape_1 (Reshape)	(None, 1, 128)	0
lstm_2 (LSTM)	(None, 64)	49,408
dense_13 (Dense)	(None, 20)	1,300

Total params: 11,219,668 (42.80 MB)
Trainable params: 11,219,668 (42.80 MB)
Non-trainable params: 0 (0.00 B)

Here's LSTM Comes with Best accuracy.

5. Model Training

The next step involves training the selected models. The CNN is trained for the task of recognizing whether a human is present in an image. For identity recognition, the LSTM is trained to handle sequential gait data and capture temporal dependencies in the walking patterns. **Transfer learning** is employed for models like VGG16 and ResNet to use pre-trained weights on ImageNet, improving their performance and reducing training time.

6. Model Evaluation

Once the models are trained, they are evaluated on a validation set to assess their performance. Metrics such as **accuracy**, **precision**, **recall**, and **F1-score** are calculated to measure the models' effectiveness in detecting humans and recognizing identities. A confusion matrix is also used to visualize the classification results and check for any misclassifications or biases.

7. Model Tuning

Based on the evaluation, the models undergo tuning to optimize performance. This includes adjusting hyperparameters such as learning rates, batch sizes, and the number of epochs. Regularization techniques like **dropout** and **batch normalization** are applied to prevent overfitting. Fine-tuning the pre-trained models, such as ResNet and VGG16, helps improve the accuracy for gait recognition.

8. Deployment and Future Work

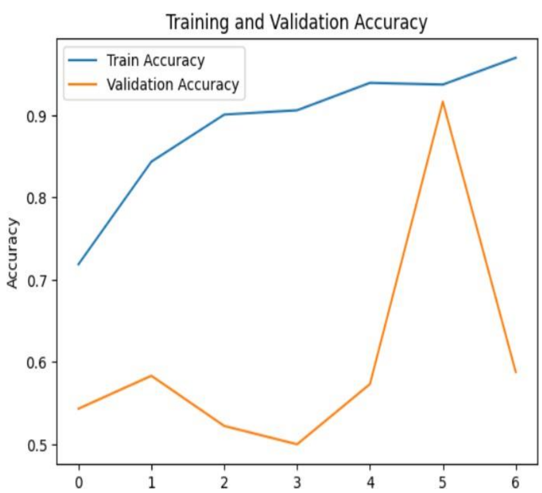
The final step involves deploying the trained models in a simulated or real-world environment. The models can be integrated into security or surveillance systems for real-time human identification through gait. Future work may involve expanding the dataset to include more diverse walking patterns, improving model efficiency, or combining multiple models to enhance robustness and accuracy in dynamic real-world applications.

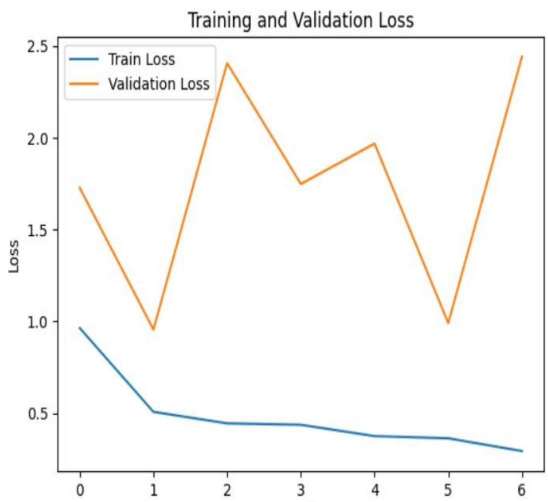
This 8-step process ensures that the gait recognition system is robust, efficient, and ready for real-world deployment.

-

Result:

```
Epoch 1/20
478/478 ━━━━━━━━━━━━━━━━━━━━ 47s 96ms/step - accuracy: 0.8964 - loss: 0.5289 - val_accuracy: 0.5725 - val_loss: 1.6914
Epoch 2/20
478/478 ━━━━━━━━━━━━━━━━━━━━ 3s 7ms/step - accuracy: 0.9375 - loss: 0.4882 - val_accuracy: 0.5833 - val_loss: 1.7122
Epoch 3/20
478/478 ━━━━━━━━━━━━━━━━━━━━ 39s 81ms/step - accuracy: 0.9317 - loss: 0.4347 - val_accuracy: 0.5583 - val_loss: 1.8338
Epoch 4/20
478/478 ━━━━━━━━━━━━━━━━━━━━ 6s 12ms/step - accuracy: 0.9375 - loss: 0.4663 - val_accuracy: 0.5833 - val_loss: 1.3713
Epoch 5/20
478/478 ━━━━━━━━━━━━━━━━━━━━ 38s 80ms/step - accuracy: 0.9521 - loss: 0.3785 - val_accuracy: 0.5620 - val_loss: 1.8915
Epoch 6/20
478/478 ━━━━━━━━━━━━━━━━━━━━ 0s 76us/step - accuracy: 0.9062 - loss: 0.4632 - val_accuracy: 0.4167 - val_loss: 1.8139
Epoch 7/20
478/478 ━━━━━━━━━━━━━━━━━━━━ 38s 80ms/step - accuracy: 0.9637 - loss: 0.3322 - val_accuracy: 0.5641 - val_loss: 1.9795
Epoch 8/20
478/478 ━━━━━━━━━━━━━━━━━━━━ 5s 11ms/step - accuracy: 0.9062 - loss: 0.3959 - val_accuracy: 0.4167 - val_loss: 1.4569
Epoch 9/20
478/478 ━━━━━━━━━━━━━━━━━━━━ 77s 80ms/step - accuracy: 0.9668 - loss: 0.3144 - val_accuracy: 0.5593 - val_loss: 2.1707
120/120 ━━━━━━━━━━━━━━━━━━━━ 6s 52ms/step - accuracy: 0.5742 - loss: 1.7872
Test Loss: 1.837316632270813
Test Accuracy: 0.5573298335075378
```





Project Link- https://github.com/Nayan4567/Gait-Recognition/tree/mainn

Conclusion

In this project, we explored two distinct tasks within the domain of gait recognition, using different methodologies and datasets.

Task 1 involved recognizing whether an image contains a human by utilizing a sample dataset. For this task, we implemented a Convolutional Neural Network (CNN) architecture that effectively captured and analysed the visual features, demonstrating reliable performance in identifying human presence. The CNN model successfully processed the dataset and provided accurate human recognition results, proving its suitability for this task.

Task 2 extended the complexity by aiming to recognize a person's identity based on their walking patterns, using the CASIA-A dataset. In this task, multiple deep learning models were employed, including CNN, ResNet, VGG16, and Long Short-Term Memory (LSTM) networks. While CNN and ResNet achieved decent results, LSTM, known for its ability to capture temporal dependencies in sequential data, outperformed the other models. The LSTM model was particularly effective in identifying subtle variations in gait, making it the most accurate model for this task. VGG16 also contributed well with its pre-trained layers, but LSTM’s sequential nature gave it an edge in dealing with the dynamic aspects of gait.

References:

[1] http://www.cbsr.ia.ac.cn/english/gait%20databases.asp

[2] https://www.sciencedirect.com/topics/computer-science/gait-recognition

[3] https://link.springer.com/article/10.1007/s41870-024-02099-z

[4] https://www.aratek.co/news/gait-recognition-biometrics-gain-attention