SIGNATURE VERIFICATION SYSTEM

DETECTING REAL AND FORGED SIGNATURES USING DEEP LEARNING AND DJANGO

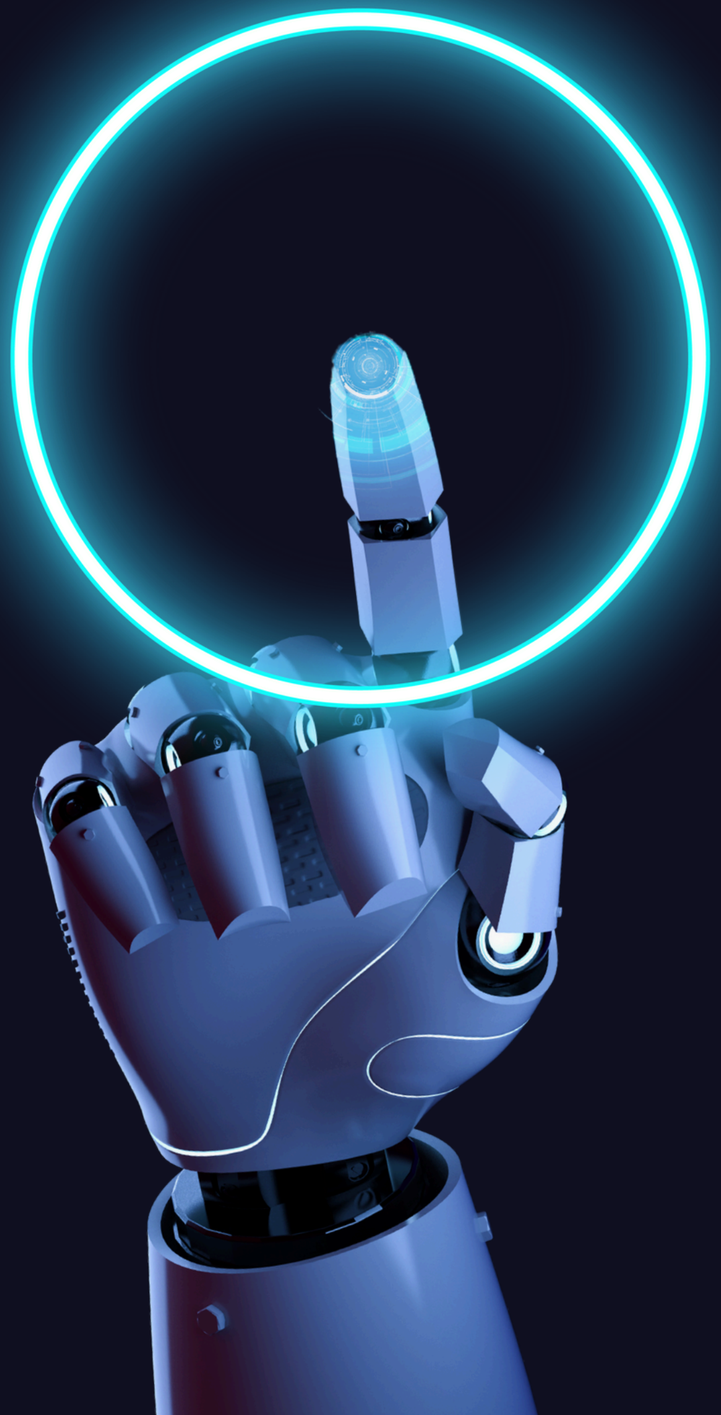
INTRODUCTION

Problem Statement:

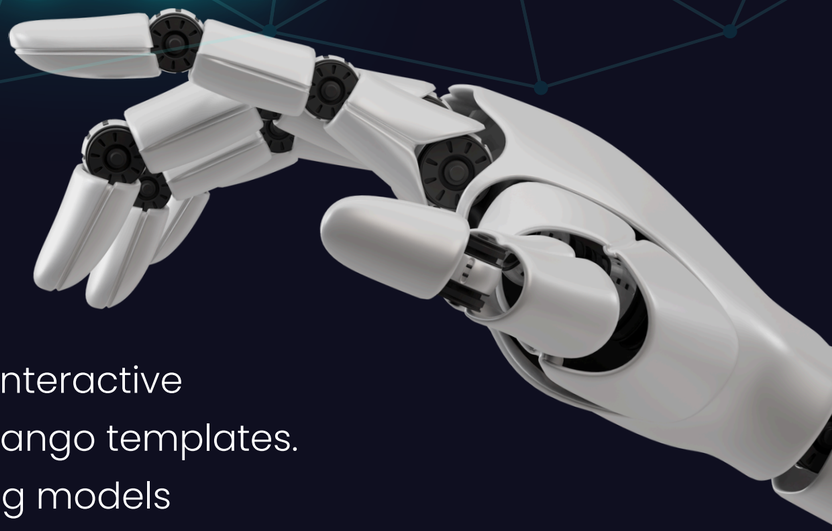
- Manual verification of signatures is time-consuming and error-prone.
- Increasing cases of document fraud demand reliable solutions.
- Need for an automated system to verify signatures with high accuracy.

Importance:

- Fraud prevention in banking and legal industries.
- Enhancing trust in digital transactions.

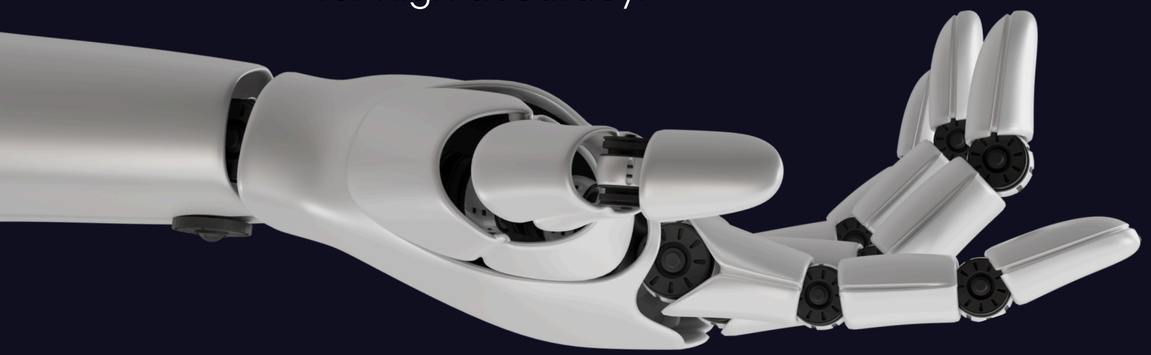


SIGNATURE VERIFICATION SYSTEM



Objective

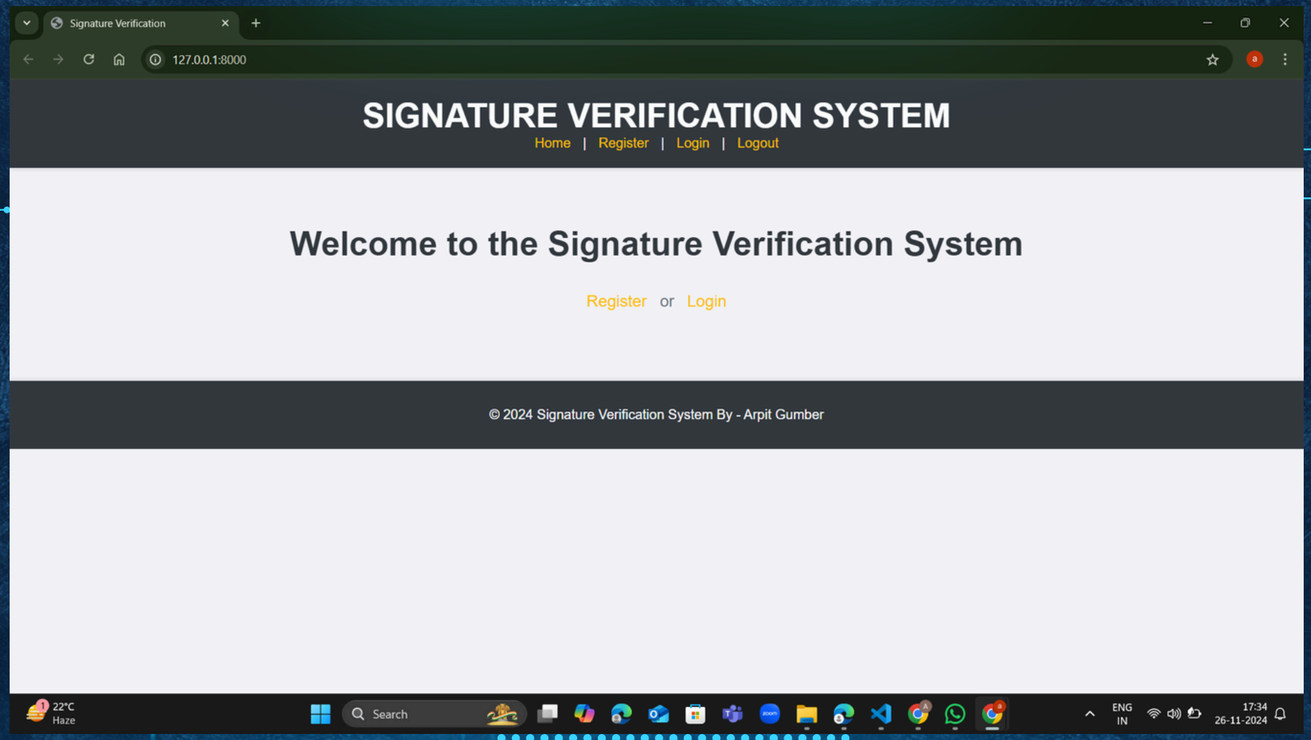
- Develop an DL-based system for accurate signature verification.
- Create a user-friendly web interface using Django.
- Incorporate multiple verification models for high accuracy.



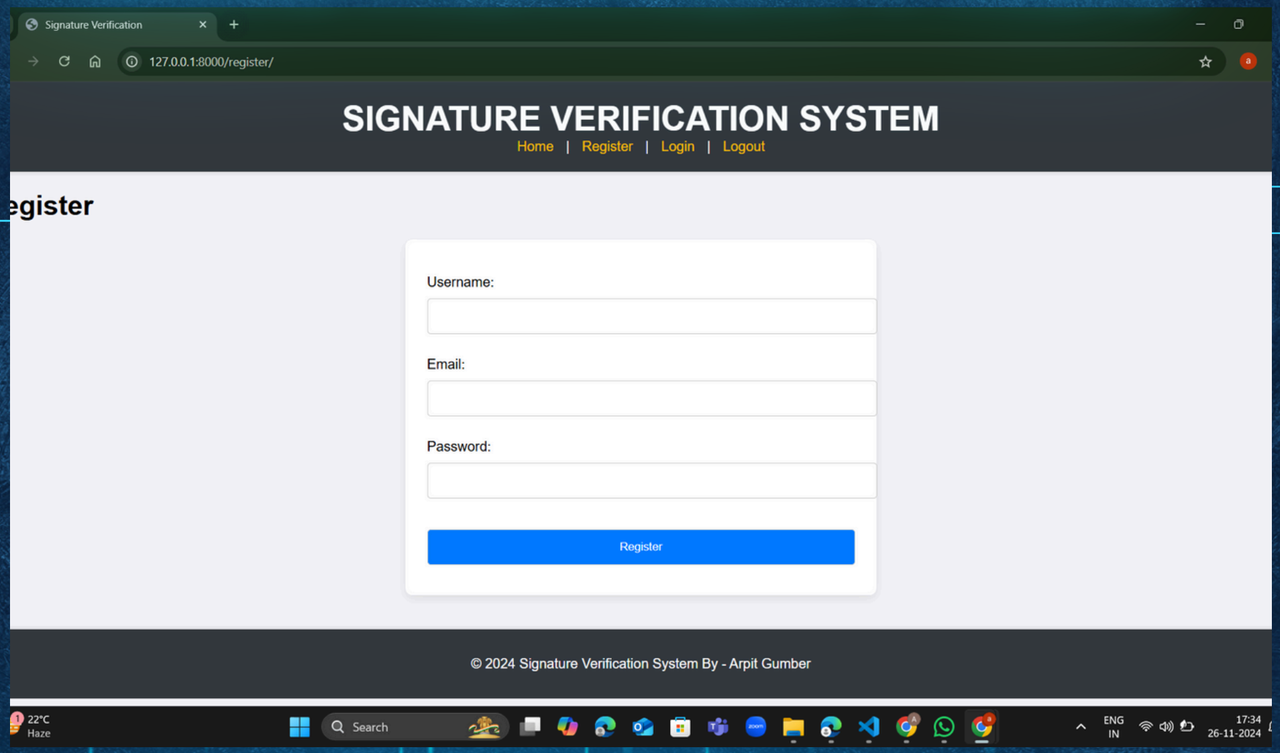
Project Overview

- Frontend: Simple and interactive interface built using Django templates.
- Backend: Deep learning models process uploaded signature images.
- Outcome:
  - Displays whether a signature is genuine or forged.
  - Includes a accuracy level for result reliability.

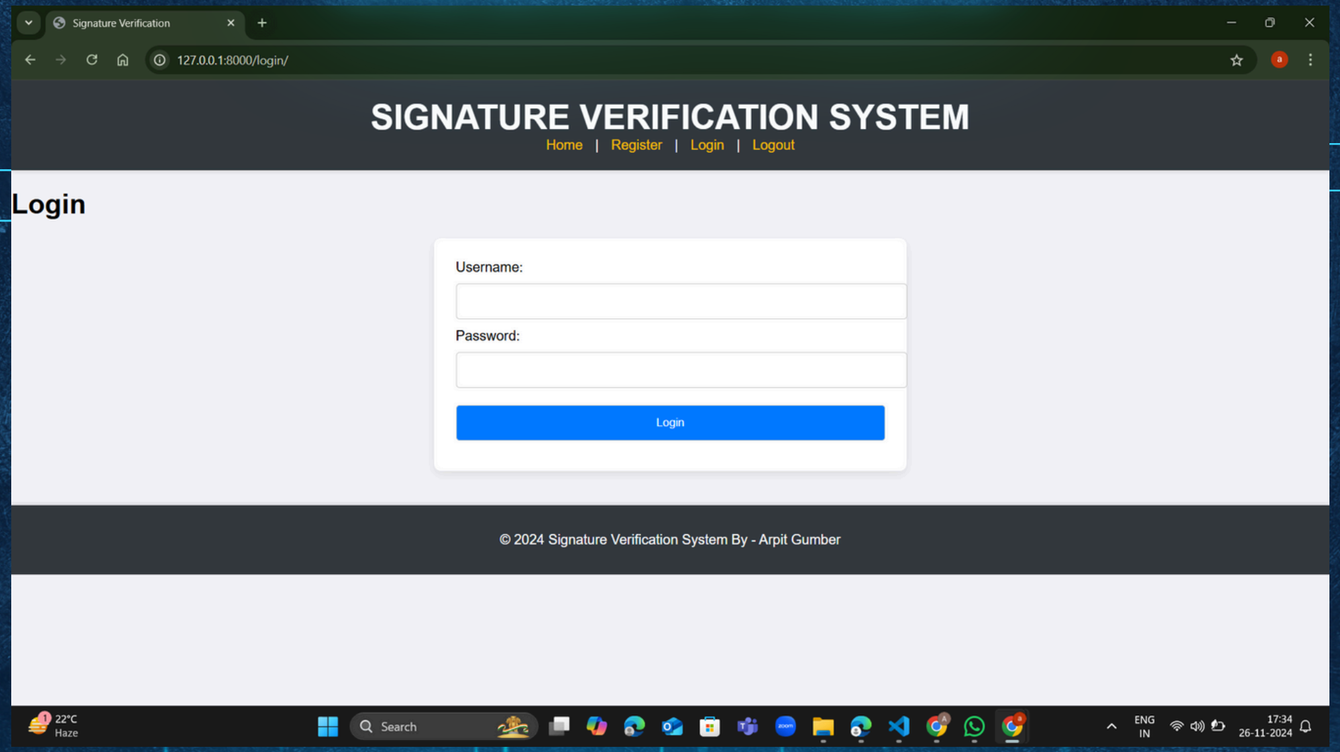
HOME PAGE


Signature Verification
127.0.0.1:8000
SIGNATURE VERIFICATION SYSTEM
Home | Register | Login | Logout
Welcome to the Signature Verification System
Register or Login
© 2024 Signature Verification System By - Arpit Gumber
22°C
Haze
Search
ENG
IN
17:34
26-11-2024

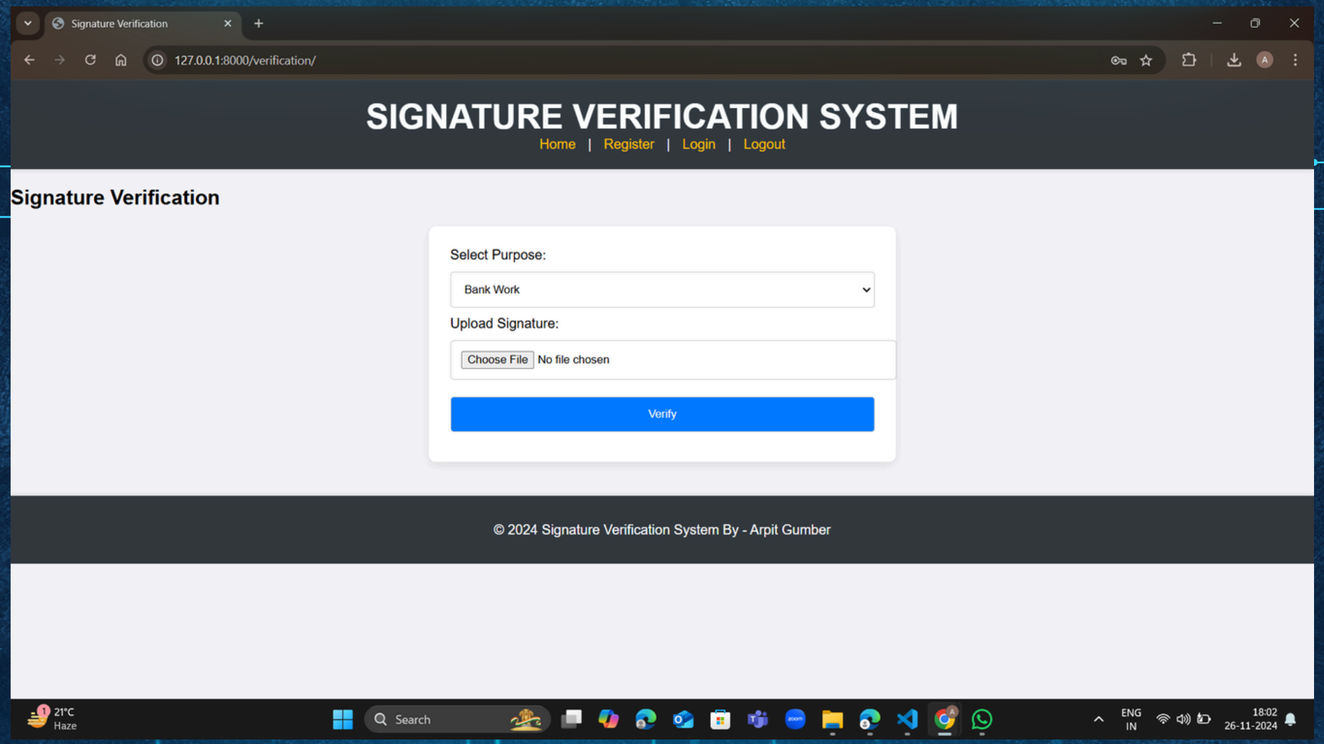
REGISTRATION PAGE



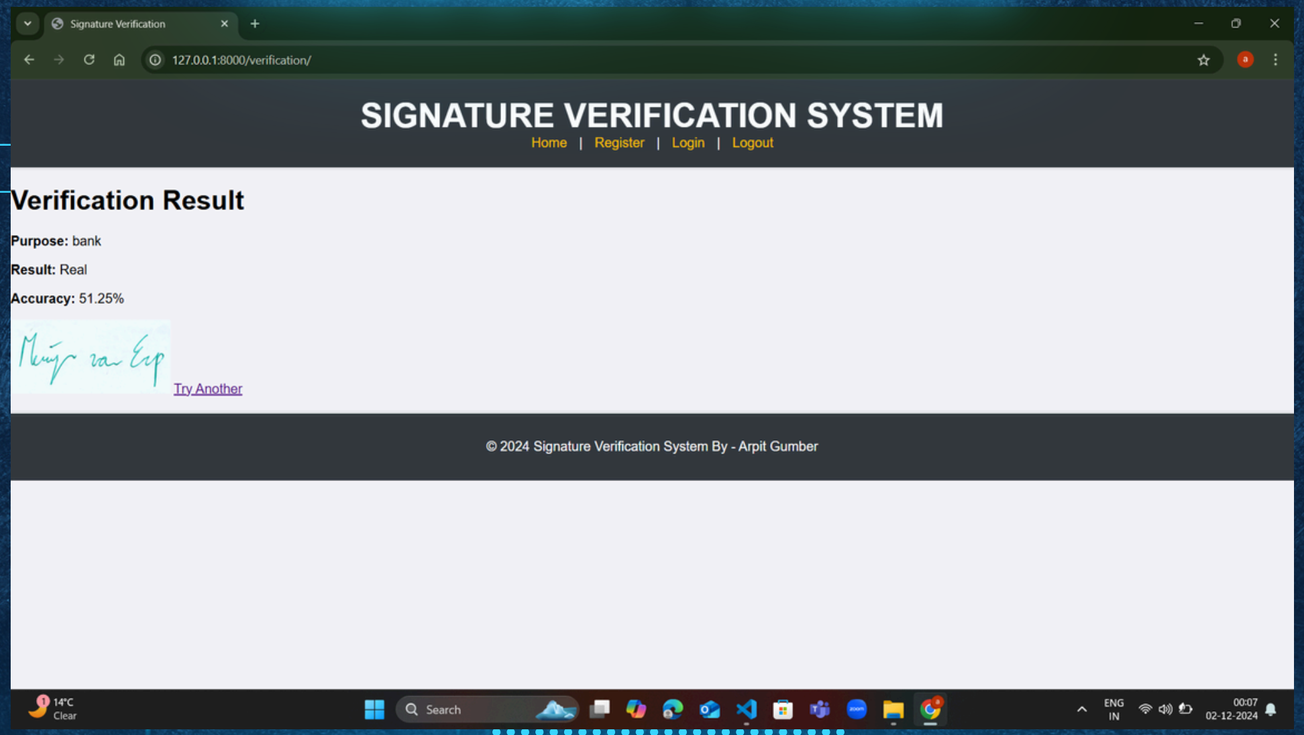
LOGIN PAGE


Signature Verification
127.0.0.1:8000/login/
SIGNATURE VERIFICATION SYSTEM
Home | Register | Login | Logout
Login
Username:
Password:
Login
© 2024 Signature Verification System By - Arpit Gumber
22°C
Haze
Search
ENG
IN
17:34
26-11-2024

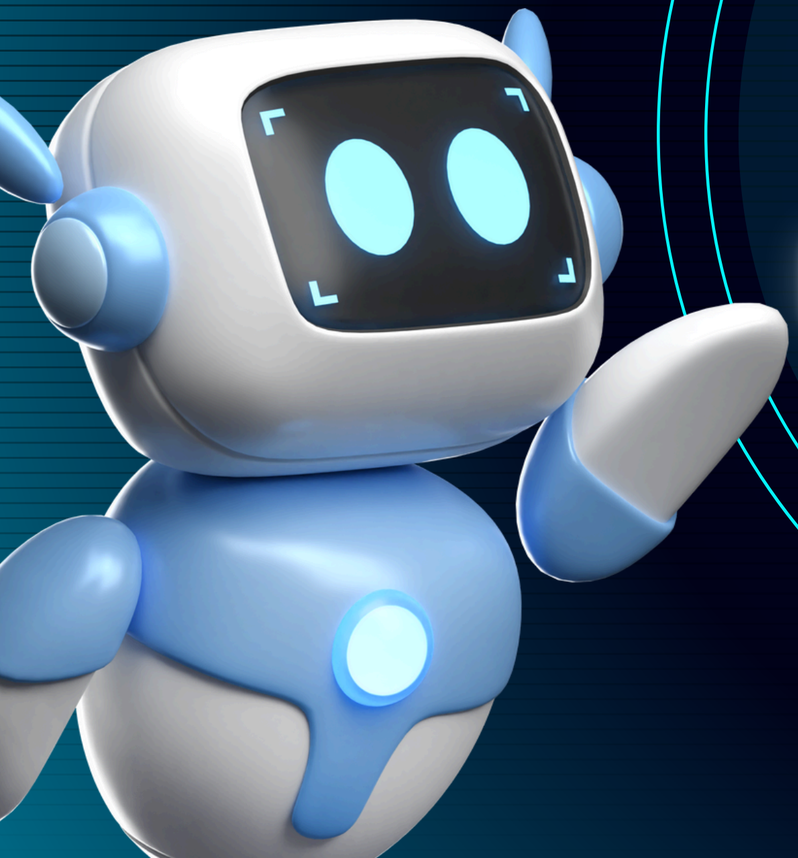
VERIFICATION PAGE



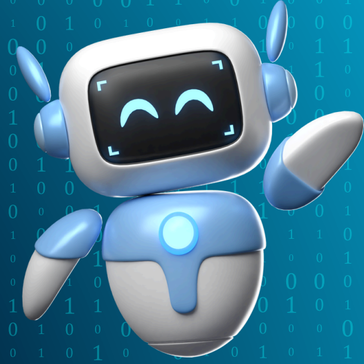
RESULT PAGE


Signature Verification
127.0.0.1:8000/verification/
SIGNATURE VERIFICATION SYSTEM
Home | Register | Login | Logout
Verification Result
Purpose: bank
Result: Real
Accuracy: 51.25%
Try Another
© 2024 Signature Verification System By - Arpit Gumber
14°C
Clear
Search
ENG
IN
00:07
02-12-2024

DEEP LEARNING MODELS USED



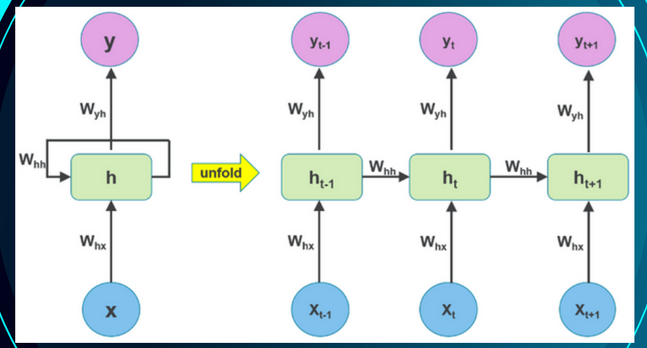
1 RECURRENT NEURAL NETWORKS (RNN)



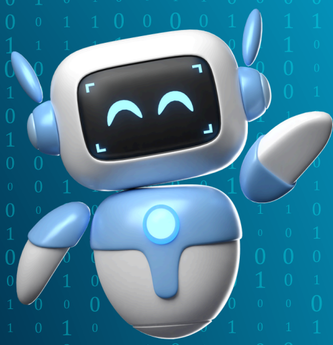
- RNNs are designed for sequential data processing and retain information from previous inputs through hidden states.
- Purpose in Project: Used to analyze the sequential nature of signature data, extracting meaningful patterns.

Working:

- Processes each step in sequence while updating hidden states.
- Suitable for time-series and sequence-related tasks like handwriting or signature analysis.
- Key Advantage: Efficient at capturing temporal dependencies.



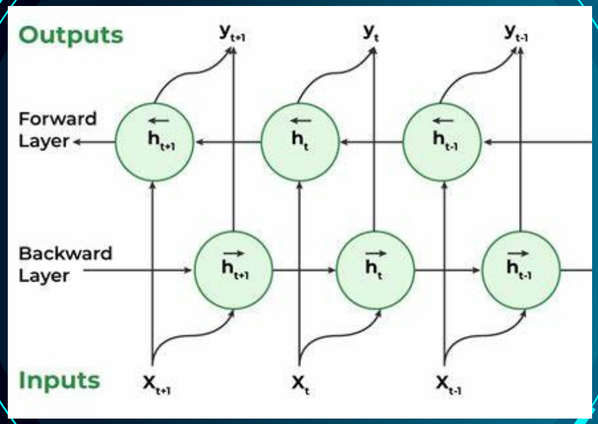
2 BIDIRECTIONAL RECURRENT NEURAL NETWORKS (BI-RNN)



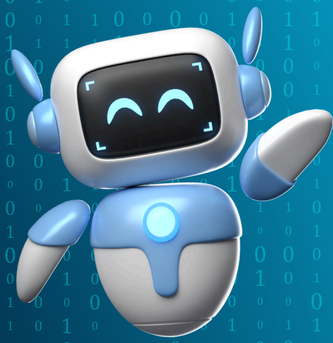
- Bi-RNNs improve on RNNs by processing data in both forward and backward directions, enhancing context understanding.
- **Purpose in Project**: Better identifies forgery traits by analyzing the signature from two perspectives (past and future contexts).

Working:

- Combines two RNNs—one processes inputs forward, while the other processes backward.
- Aggregates information from both directions to improve accuracy.
- **Key Advantage:** Provides a comprehensive understanding of sequential data.



3 CONVOLUTIONAL RECURRENT NEURAL NETWORKS (CRNN)



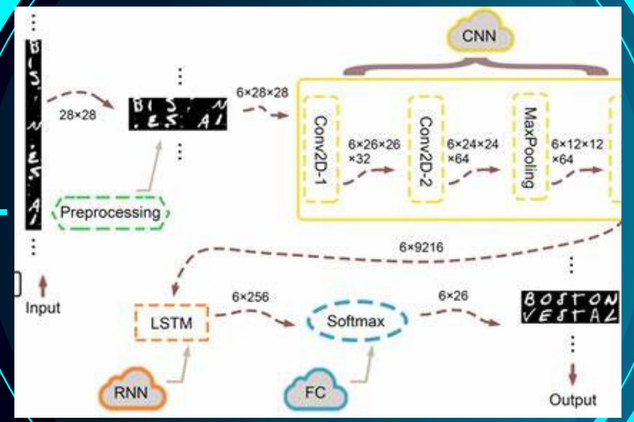
- CRNNs combine the feature extraction power of CNNs with the sequential data processing of RNNs.

Purpose in Project:

- CNN Component: OpenCV is used for image preprocessing and extracting spatial features from signatures (e.g., edges, curves).
- RNN Component: LSTM layers are used to analyze the sequential patterns of extracted features.

Working:

- The CNN extracts spatial features from the input signature image.
- These features are fed into the RNN (LSTM) to model temporal relationships and predict if the signature is real or forged.
- **Key Advantage**: Combines strengths of both architectures, making it suitable for signature verification tasks.



Accuracy Levels of Deep Learning Models

01 RNN

```
#get final loss and accuracy
final_loss,final_accuracy=model.evaluate(test_data,test_labels)

#print results in percentage form
print("final loss: {:.2f}%".format(final_loss*100))
print("final accuracy: {:.2f}%".format(final_accuracy*100))
```

```
313/313 ━━━━━━━━━━━━━━━━━━━━ 2s 7ms/step - accuracy: 0.9732 - loss: 0.0871
final loss: 6.84%
final accuracy: 97.87%
```

02 BI-RNN

```
#get the final loss and accuracy
final_loss, final_accuracy= model.evaluate(X_test, y_test)
#print the results in percentage form
print('Final loss: {:.2f}%'.format(final_loss * 100))
print('Final Accuracy: {:.2f}%'.format(final_accuracy* 100))
```

```
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 15ms/step - accuracy: 0.5019 - loss: 0.6927
Final loss: 69.27%
Final Accuracy: 50.69%
```

03 CRNN

```
#get the final loss and accuracy
final_loss, final_accuracy = model.evaluate(test_ds)

#print the results in percentage form
print('Final loss: {:.2f}%'.format(final_loss * 100))
print('Final Accuracy: {:.2f}%'.format(final_accuracy* 100))
```

```
3/3 ━━━━━━━━━━━━━━━━━━━━ 1s 328ms/step - accuracy: 0.5115 - loss: 0.6933
Final loss: 69.39%
Final Accuracy: 51.25%
```

DJANGO

WHY DJANGO FOR WEB APPLICATIONS?

- Simplifies building complex applications with features like user authentication, database management, and URL routing.
- Scalable for both small and large projects.

ROLE OF DJANGO IN OUR PROJECT

- **User Authentication:** Django's built-in authentication system manages login and registration seamlessly.
- **File Handling:** Allows users to upload signature images securely for verification.
- **Backend Logic:** Connects the frontend interface with deep learning models for signature verification.
- **Routing**: Manages smooth navigation between login, registration, and upload pages.

GIT & GITHUB

GIT

- Git is a distributed version control system used to track changes in source code during software development.
- It enables collaboration by allowing multiple developers to work on the same project simultaneously and merge changes effectively.
- With features like branching, merging, and version tracking, Git ensures a flexible and robust workflow.

GITHUB

- GitHub is a cloud-based platform for hosting and managing Git repositories.
- It provides tools for collaboration, such as pull requests, code reviews, and issue tracking, making team development more streamlined.

THANK YOU!