A HYBRID METHOD OF FEATURE EXTRACTION FOR SIGNATURES VERIFICATION USING CNN AND HOG A MULTI-CLASSIFICATION APPROACH

Syed Viquar Uddin Hasan[1], Mohammed Abdul Salam[2], Syeed Mohammed Faisal[3], Neha Naznein[4]

[1,2,3]B. E Student, Department of IT, ISL College of Engineering, India.

[4]Assistant Professor, Department of IT, ISL College of Engineering, Hyderabad, India.

syedmohiahmed3@gmail.com, omairmohammed277@gmail.com, syedfaisal2m1786@gmail.com

ABSTRACT

The quantity and calibration of the extracted features determine how well these systems can distinguish between authentic and fake signatures, making the feature extraction phase of an offline signature verification system critical to the overall performance of these systems.The study presents a hybrid technique for offline signature verification systems feature extraction from signature pictures. To find important features, the approach combines the usage of Convolutional Neural Network (CNN) and Histogram of Oriented Gradients (HOG) approaches. A decision tree feature selection algorithm then comes into play. The hybrid approach was tested on two datasets (UTSig and CEDAR) and assessed using three classifiers: K-nearest Neighbor, support vector machine, and long short-term memory. Even for expert forgeries, the testing findings demonstrated a high degree of accuracy in identifying genuine from fabricated signatures.

INTRODUCTION

The most significant technical approach for identifying individuals and assessing their power based on their unique physiology and behavior is biometrics. One of the biometric verification techniques that is most widely recognized worldwide is the handwritten signature. Handwritten signatures are used as distinct behavioral biometrics in financial documents, credit cards, passports, banks, and check processing. These signatures are hard to verify, especially if they're not explicit. To reduce the likelihood of theft or fraud, a system that can differentiate between a real signature and a false signature is necessary. Many research in this area have been carried out over the last thirty years, moving from machine learning algorithms to deep learning algorithms, from conventional verification based on expert judgments. Even after all these research, offline signature verification systems still need a great deal of work.In light of the conversation, the current study attempts to create a hybrid approach that uses machine learning and deep learning classifiers to distinguish between genuine and fake signatures in signature images. This hybrid approach should also be able to improve the performance of different classifiers. We'll verify signatures offline using this technique.

PROBLEM STATEMENT

The problem addressed in this study is the need for robust offline signature verification systems. Existing systems often struggle to effectively differentiate between genuine and forged signatures, especially when

dealing with skilled forgeries. This research aims to enhance system performance by introducing a hybrid feature extraction method and evaluating its effectiveness using various classifiers on different datasets.

LITERATURE SURVEY

Signature Forgery Detection Using Machine Learning:

[PDF] SIGNATURE FORGERY DETECTION USING MACHINE LEARNING | Semantic Scholar

ABSTRACT: In today's society, signature are used many important documents such bank cheque, passport, driving license, etc. and can be faked in multiple ways. This creating many problems such as fake identifications, identify theft, hacking etc. To reduce this issue, our project is focused on developing a system for detecting whether a signature is real or fake from dataset of signatures using CNN and Deep learning. The reason we are using CNN and deep learning is because signature change over a period of time based on multiple behavioral changes such age, state of mind, physical health etc. We require a system that can learn from multiple training datasets and increase its accuracy of detection. There are two types of signatures authentication methods, which are online signature and offline signature verification methods. Our project is based on offline signatures forgery detection method .This type of signatures is handwritten on the documents and require an image of the signature. This is why we also should consider image processing for this project. We are referencing a few papers which implement the project using a few methods for both online and offline signature forgery detection methods based on deep learning models, we plan on implementing the offline methods and try to achieve a better accuracy.

Handwritten Signatures Forgery Detection Using Pre-Trained Deep Learning Methods:

(PDF) HANDWRITTEN SIGNATURES FORGERY DETECTION USING PRE-TRAINED DEEP LEARNING METHODS (researchgate.net)

ABSTRACT: Handwritten signature recognition (HSR) is crucial in various applications, such as document verification, authentication, financial transactions, banking transactions, and legal agreements. However, the prevalence of signature forgery poses a significant challenge to the integrity and security of these authentication systems. The purpose of signature forgery detection (SFD) systems is to discriminate between genuine signatures (by the purported person) and forged ones (by an impostor), which is a challenging task, especially in offline scenarios that use scanned signature images for signature recognition, where dynamic information about the signing process is not available. In recent years, pre-trained deep learning (DL) models have been widely used in image processing tasks due to their ability to achieve high accuracy with minimal training time and computational resources. By leveraging pre-trained models, developers can avoid starting from scratch when training a model, which can save time. Therefore, some pre-trained DL models are used for SFD in this paper and compared with each other. The result of implementing these methods shows that these methods have good accuracy for SFD. The MobileNet model, in particular, shows remarkable accuracy, reaching approximately 98.44%. In addition, it offers the advantages of relatively short training time and compact model size. These valuable features make MobileNet suitable for deploying mobile devices and embedded systems.

An integrated approach on verification of signatures using multiple classifiers (SVM and Decision Tree): A multi-classification approach:

http://www.science-gate.com/IJAAS/2022/V9I1/1021833ijaas202201012.html

ABSTRACT: A signature is a handwritten representation that is commonly used to validate and recognize the writer individually. An automated verification system is mandatory to verify the identity. The signature essentially displays a variety of dynamics and the static characteristics differ with time and place. Many scientists have already found different algorithms to boost the signature verification system function extraction point. The paper is aimed at multiplying two different ways to solve the problem in digital, manual, or some other means of verifying signatures. The various characteristics of the signature were found through the most adequately implemented methods of machine learning (support vector and decision tree). In addition, the characteristics were listed after measuring the effects. An experiment was performed in various language databases. More precision was obtained from the feature.

Recent developments in pretreatment technologies on lignocellulosic biomass: Effect of key parameters, technological improvements, and challenges:

Recent developments in pretreatment technologies on lignocellulosic biomass: Effect of key parameters, technological improvements, and challenges - ScienceDirect

ABSTRACT: Lignocellulosic biomass is an inexpensive renewable source that can be used to produce biofuels and bioproducts. The recalcitrance nature of biomass hampers polysaccharide accessibility for enzymes and microbes. Several pretreatment methods have been developed for the conversion of lignocellulosic biomass into value-added products. However, these pretreatment methods also produce a wide range of secondary compounds, which are inhibitory to enzymes and microorganisms. The selection of an effective and efficient pretreatment method discussed in the review and its process optimization can significantly reduce the production of inhibitory compounds and may lead to enhanced production of fermentable sugars and biochemicals. Moreover, evolutionary and genetic engineering approaches are being used for the improvement of microbial tolerance towards inhibitors. Advancements in pretreatment and detoxification technologies may help to increase the productivity of lignocellulose-based biorefinery. In this review, we discuss the recent advancements in lignocellulosic biomass pretreatment technologies and strategies for the removal of inhibitors.

Offline Handwritten Signature Verification Using Deep Neural Networks:

Offline Handwritten Signature Verification Using Deep Neural Networks | Semantic Scholar

ABSTRACT: Prior to the implementation of digitisation processes, the handwritten signature in an attendance sheet was the preferred way to prove the presence of each student in a classroom. The method is still preferred, for example, for short courses or places where other methods are not implemented. However, human verification of handwritten signatures is a tedious process. The present work describes two methods for classifying signatures in an attendance sheet as valid or not. One method based on Optical Mark Recognition is general but determines only the presence or absence of a signature. The other method uses a multiclass convolutional neural network inspired by the AlexNet architecture and, after training with a few pieces of genuine training data, shows over 85% of precision and recall recognizing the author of the signatures. The use of data augmentation and a larger number of genuine signatures ensures higher accuracy in validating the signatures.

SYSTEM ARCHITECTURE:

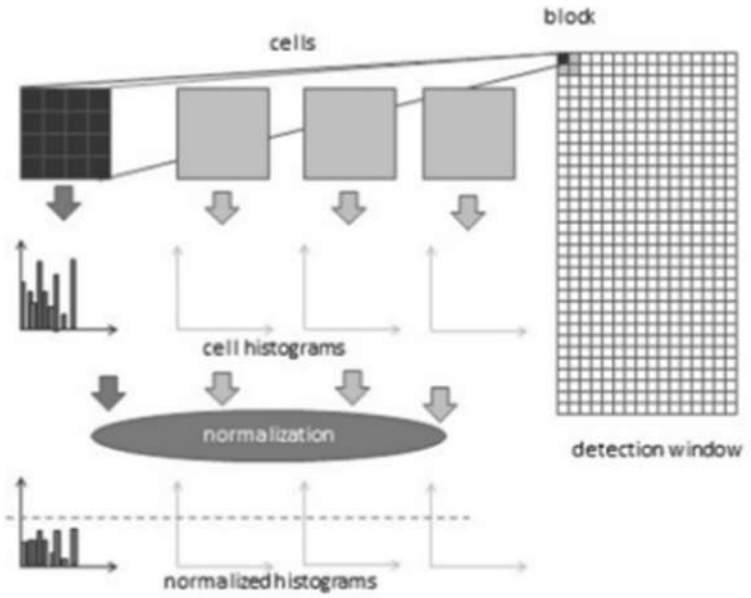


Fig.1 System architecture

DATA FLOW DIAGRAM:

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
4. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

IMPLEMENTATION

MODULES:

- Data exploration: using this module we will load data into system
- Processing: Using the module we will read data for processing
- Splitting data into train & test: using this module data will be divided into train & test
- Model generation: Model building - CNN, Feature Extraction using HOG, Feature Extraction using CNN and HOG with Feature Selection using DT with RFE, SVM, KNN, LSTM, Voting Classifier (RF + DT)
- User signup & login: Using this module will get registration and login

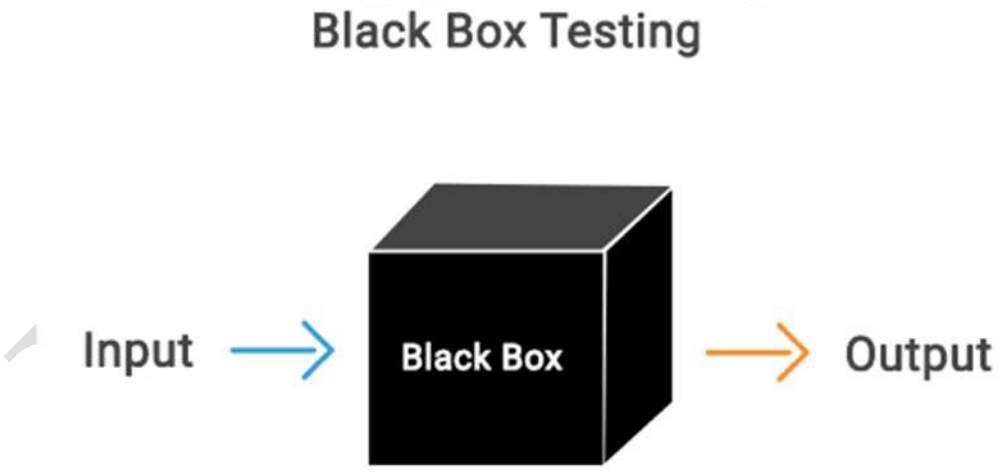
- User input: Using this module will give input for prediction
- Prediction: final predicted displayed

SYSTEM TESTING

System testing, also referred to as system-level tests or system-integration testing, is the process in which a quality assurance (QA) team evaluates how the various components of an application interact together in the full, integrated system or application. System testing verifies that an application performs tasks as designed. This step, a kind of black box testing, focuses on the functionality of an application. System testing, for example, might check that every kind of user input produces the intended output across the application.

Behavioral Testing:

The final stage of testing focuses on the software's reactions to various activities rather than on the mechanisms behind these reactions. In other words, behavioral testing, also known as black-box testing, presupposes running numerous tests, mostly manual, to see the product from the user's point of view. QA engineers usually have some specific information about a business or other purposes of the software ('the black box') to run usability tests, for example, and react to bugs as regular users of the product will do. Behavioral testing also may include automation (regression tests) to eliminate human error if repetitive activities are required. For example, you may need to fill 100 registration forms on the website to see how the product copes with such an activity, so the automation of this test is preferable.



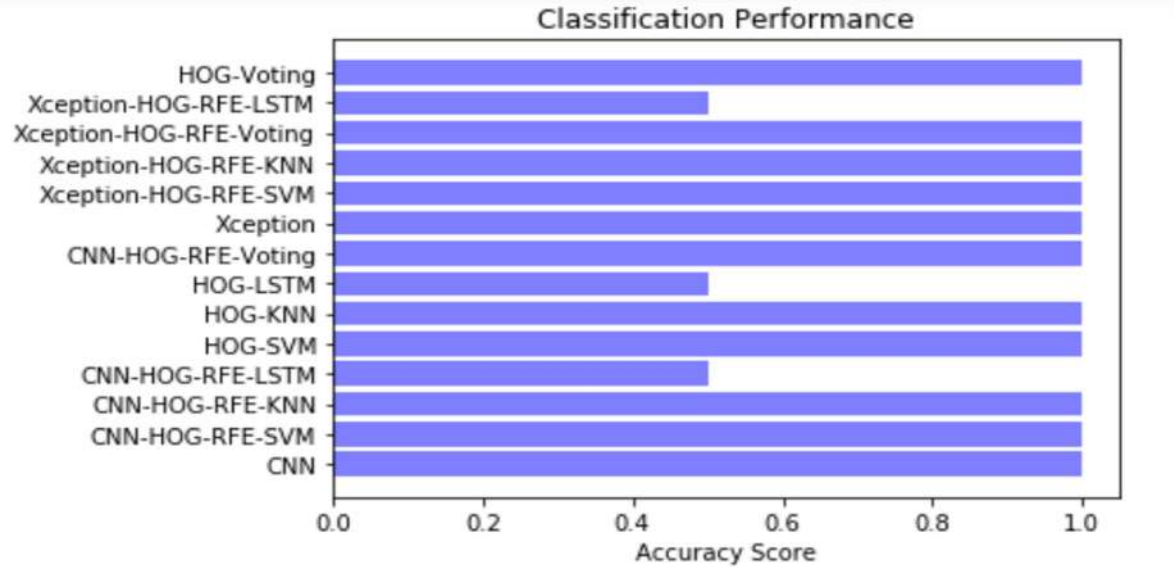
TEST CASES:

S.NO	INPUT	If available	If not available
1	User signup	User get registered into the application	There is no process

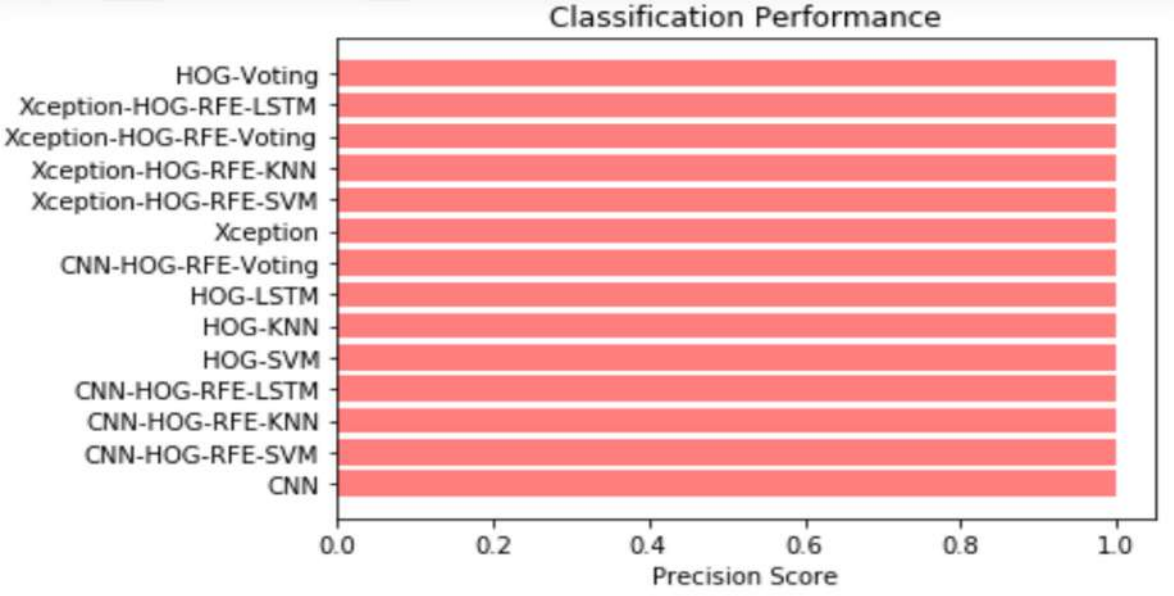
2	User sign in	User get login into the application	There is no process
3	Enter input for prediction	Prediction result displayed	There is no process

RESULTS

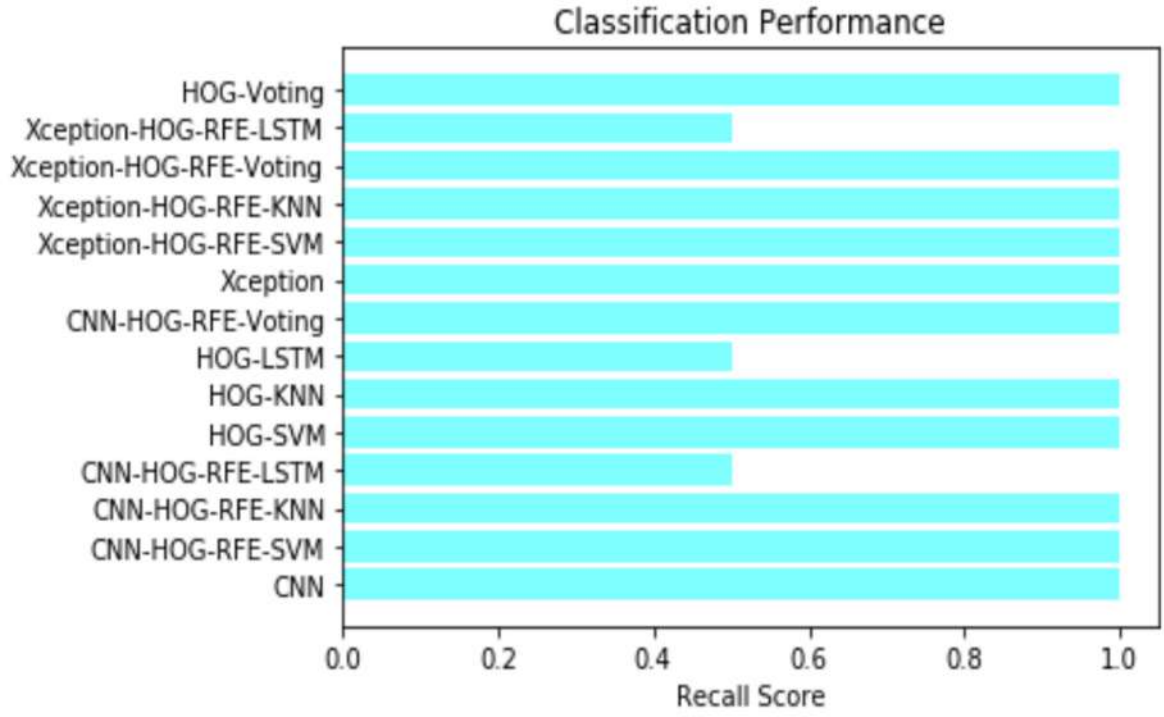
SCREENS:



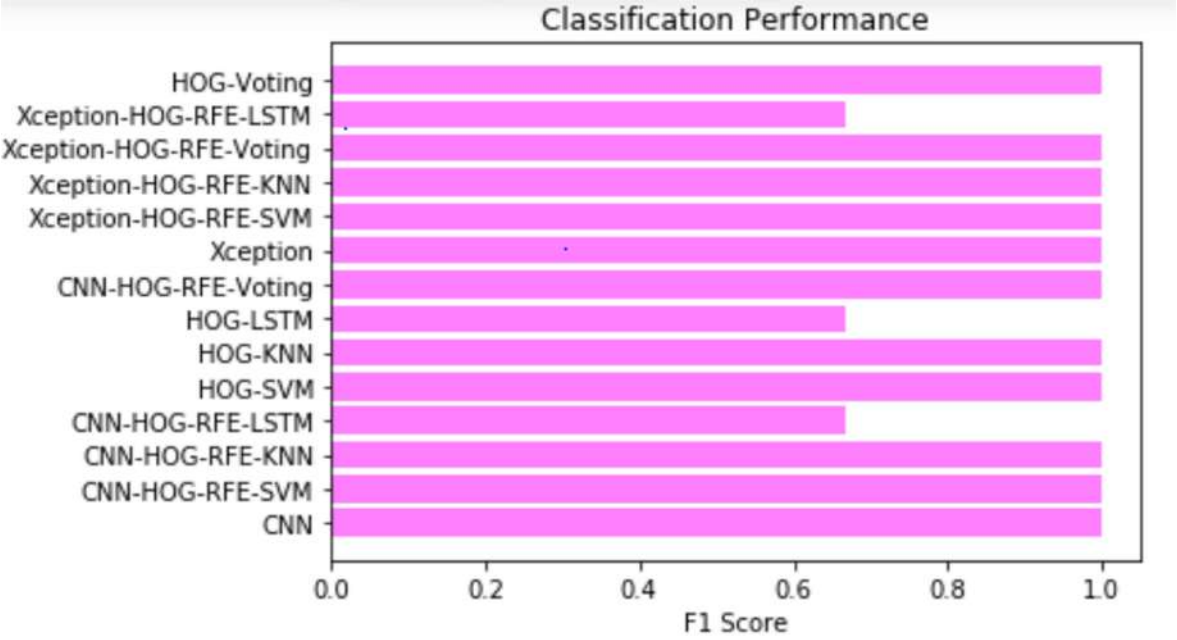
ACCURACY COMPARISION GRAPH - CEDAR DATASET



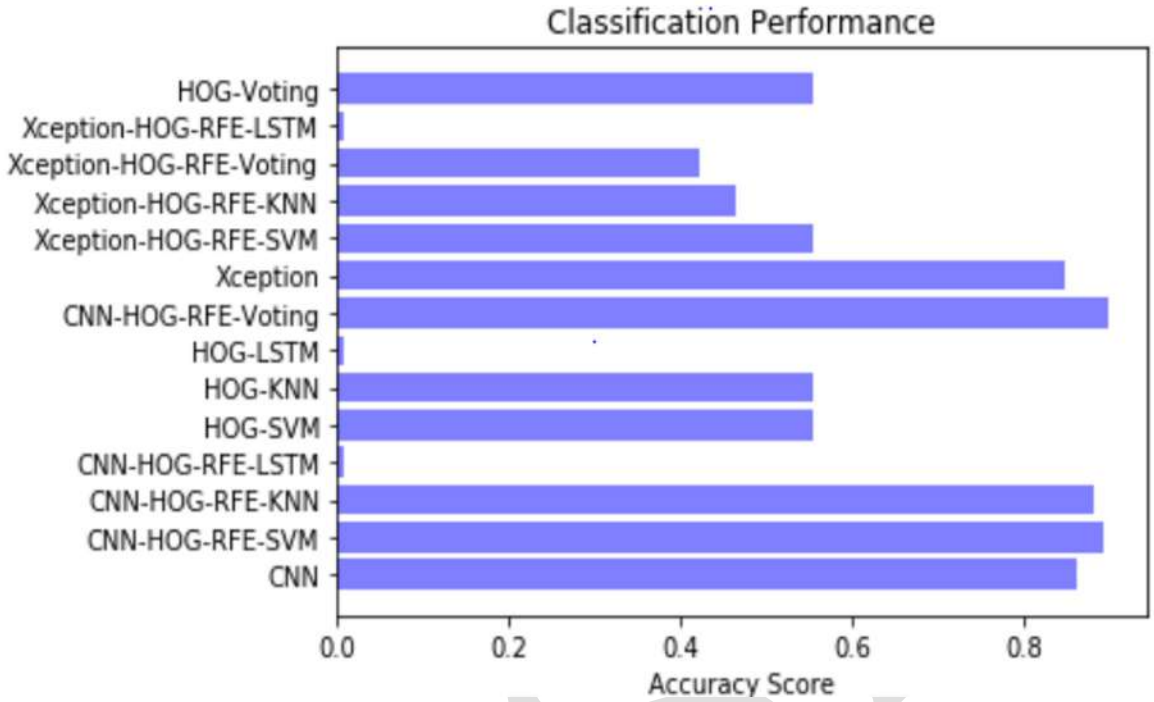
PRECISION COMPARISION GRAPH - CEDAR DATASET



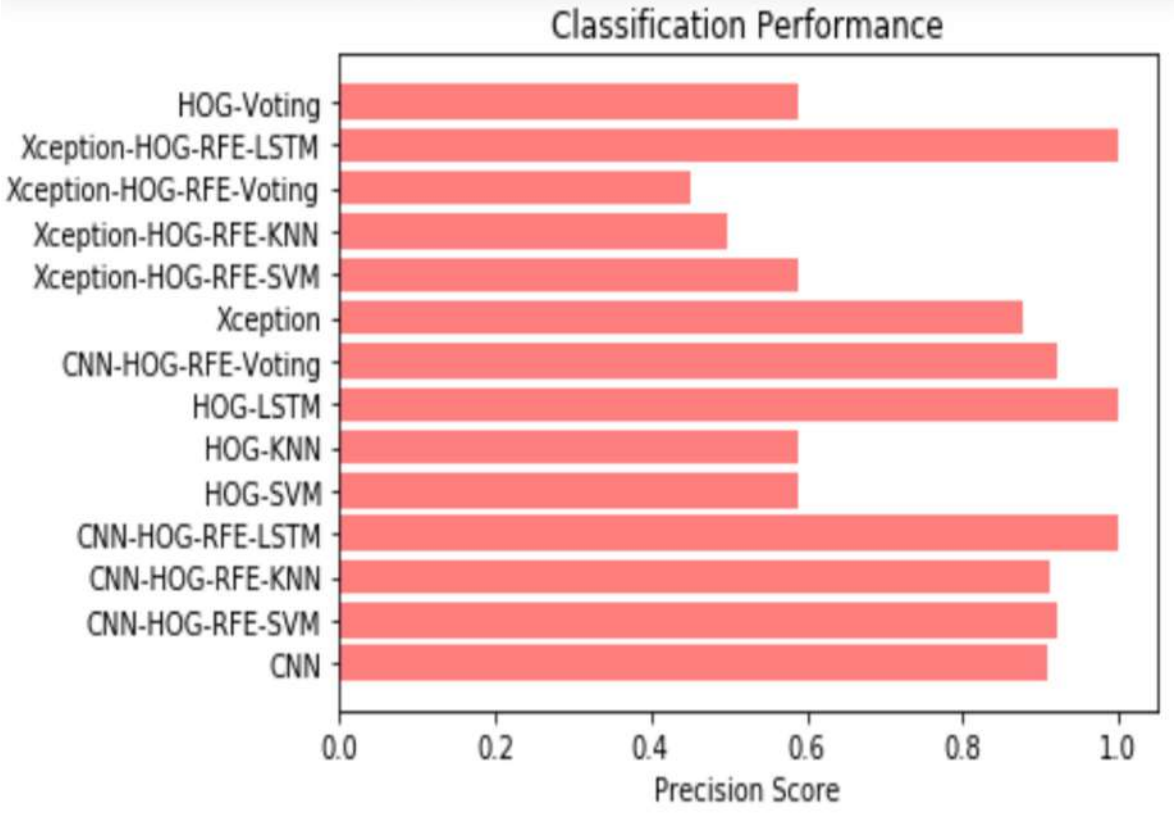
RECALL COMPARISION GRAPH - CEDAR DATASET



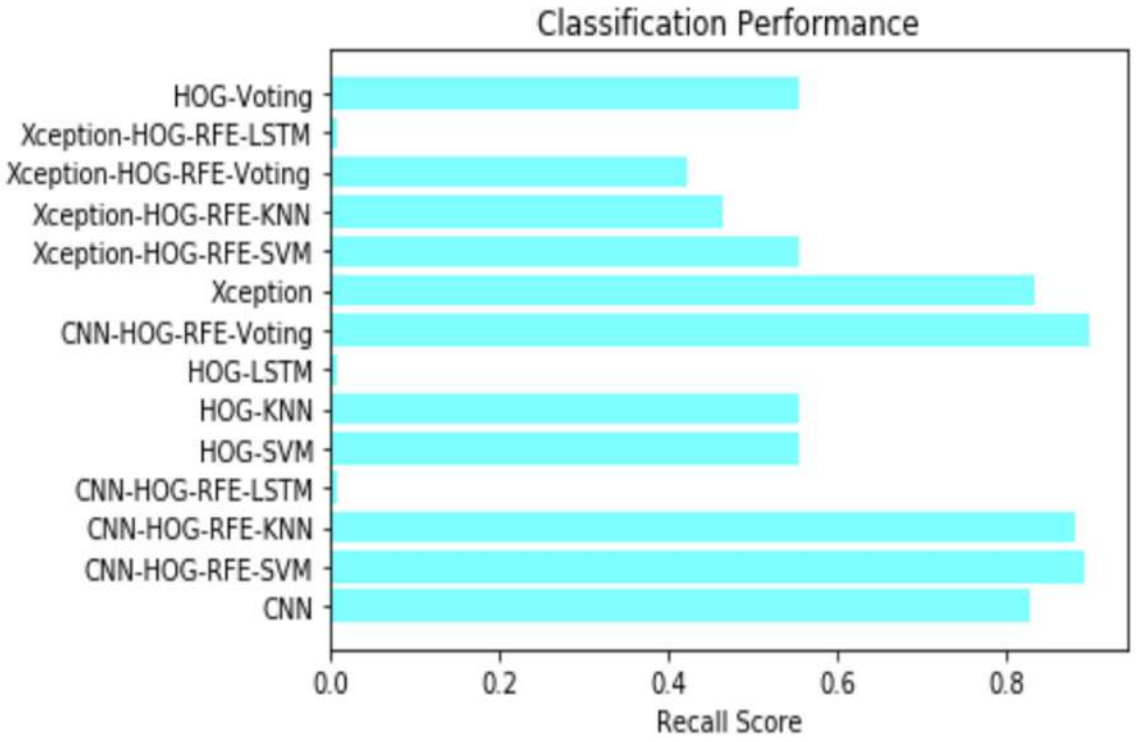
F1 SCOPE COMPARISION GRAPH - CEDAR DATASET



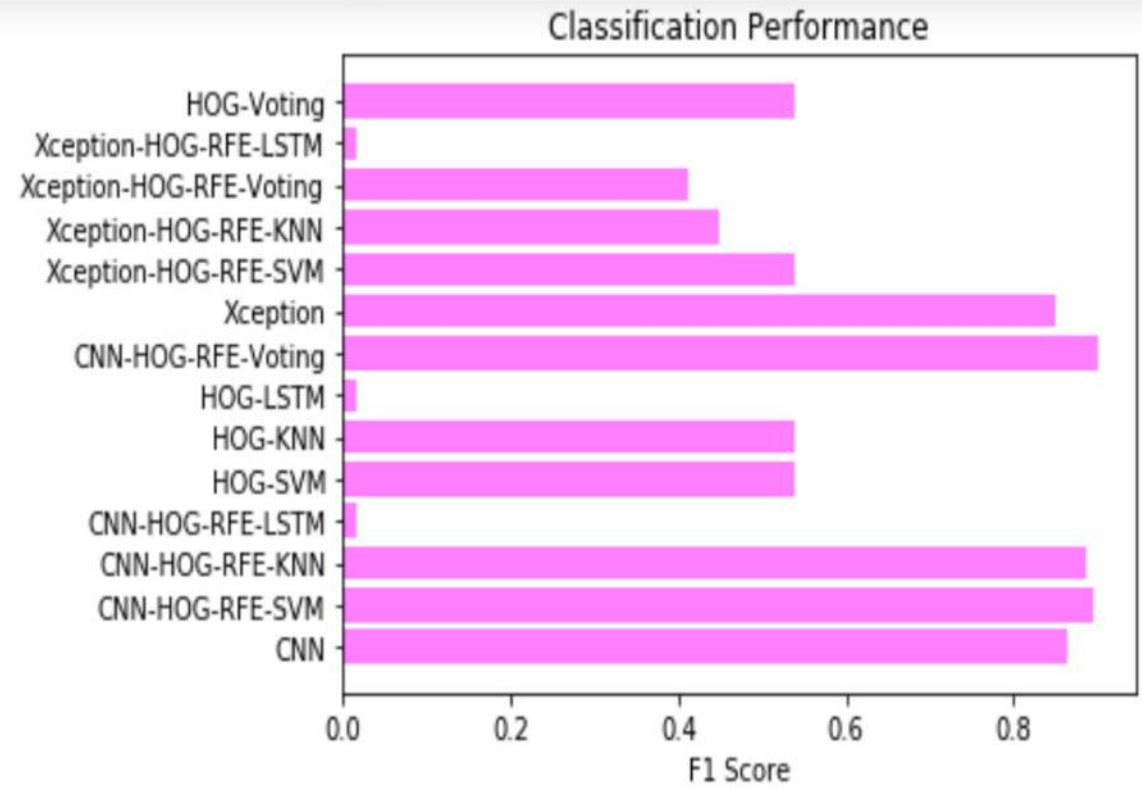
ACCURACY COMPARISION GRAPH - UTSig DATASET



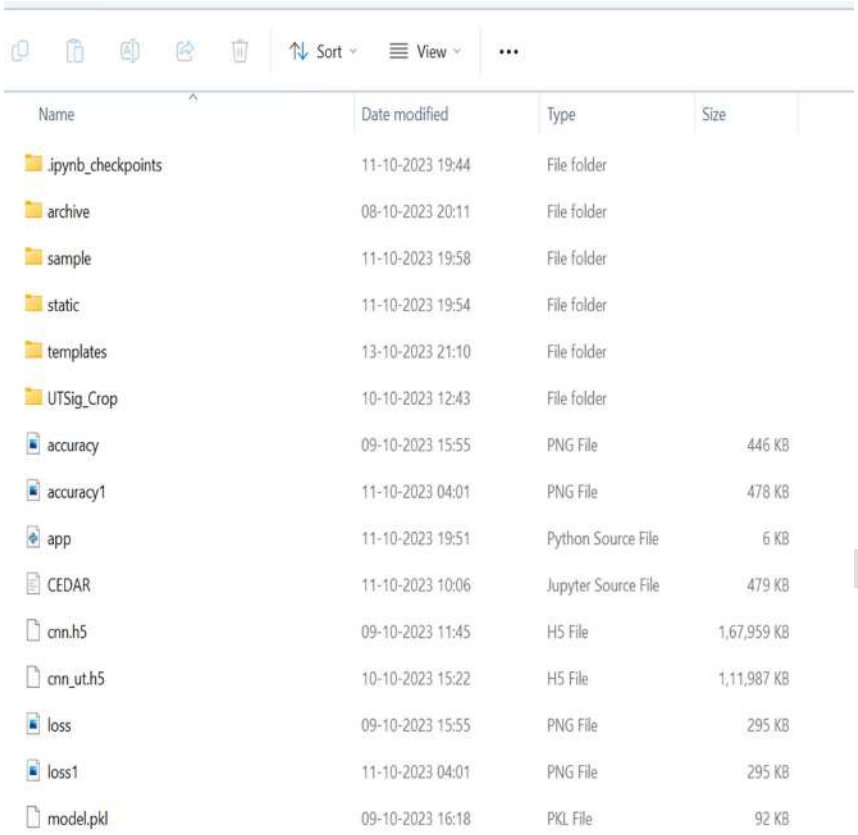
PRECISION COMPARISION GRAPH - UTSig DATASET

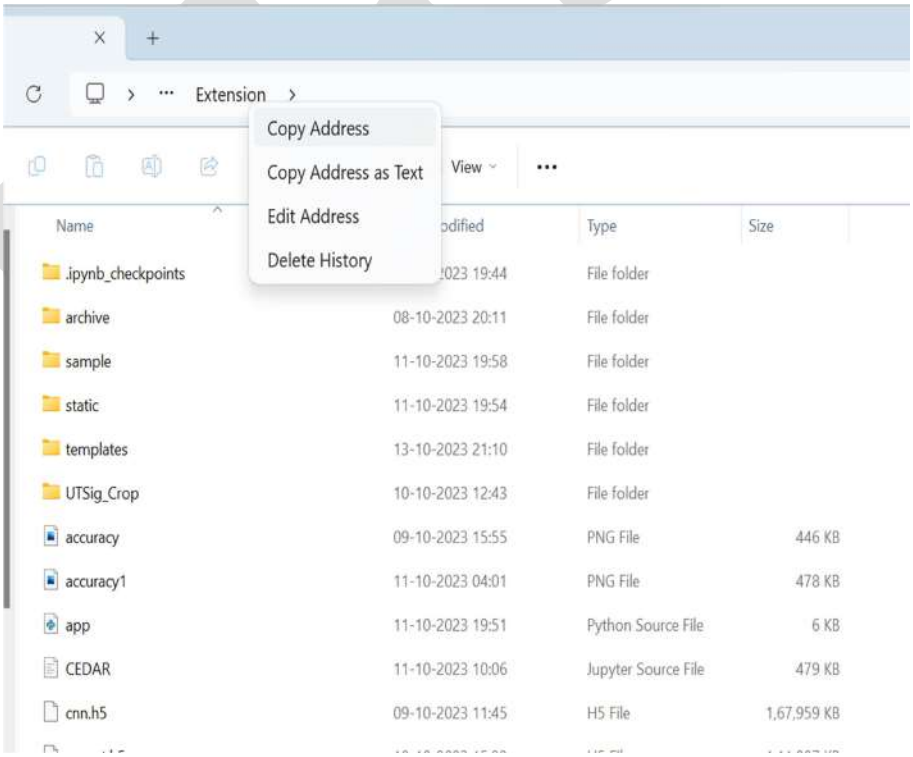


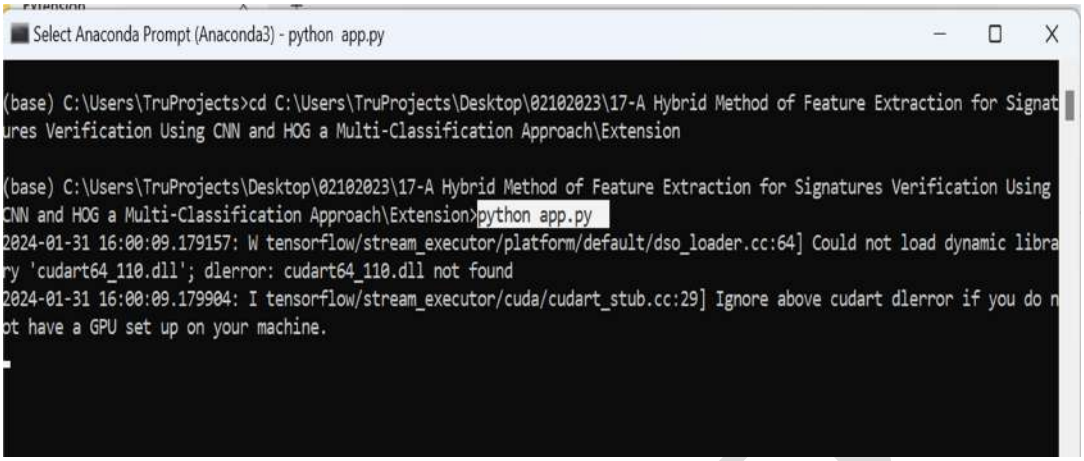
RECALL COMPARISION GRAPH - UTSig DATASET

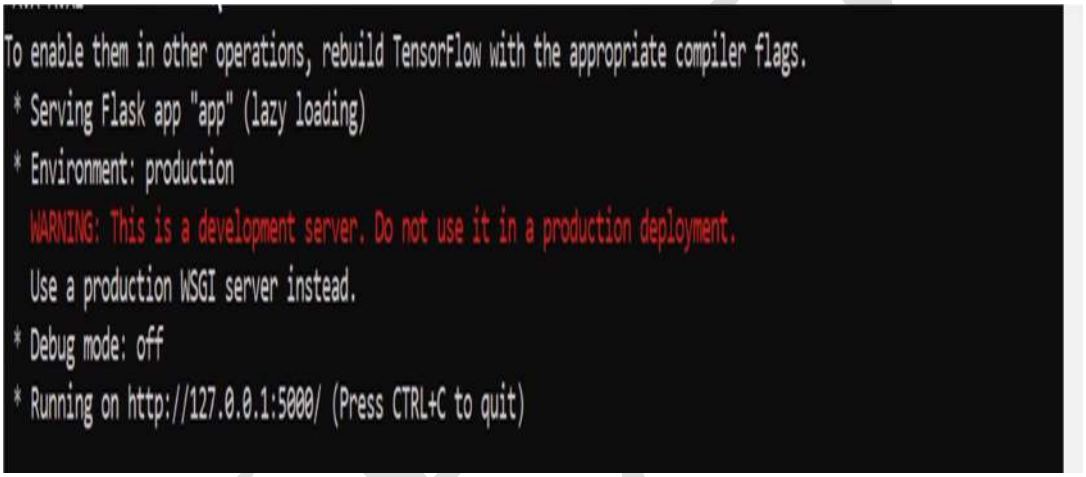


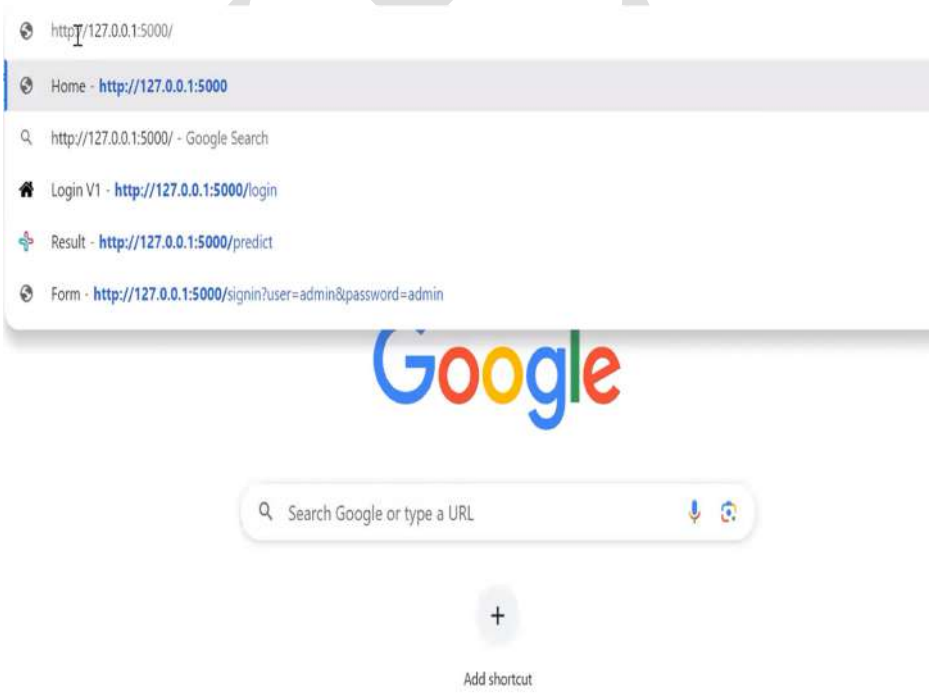
F1 SCORE COMPARISION GRAPH - UTSig DATASET

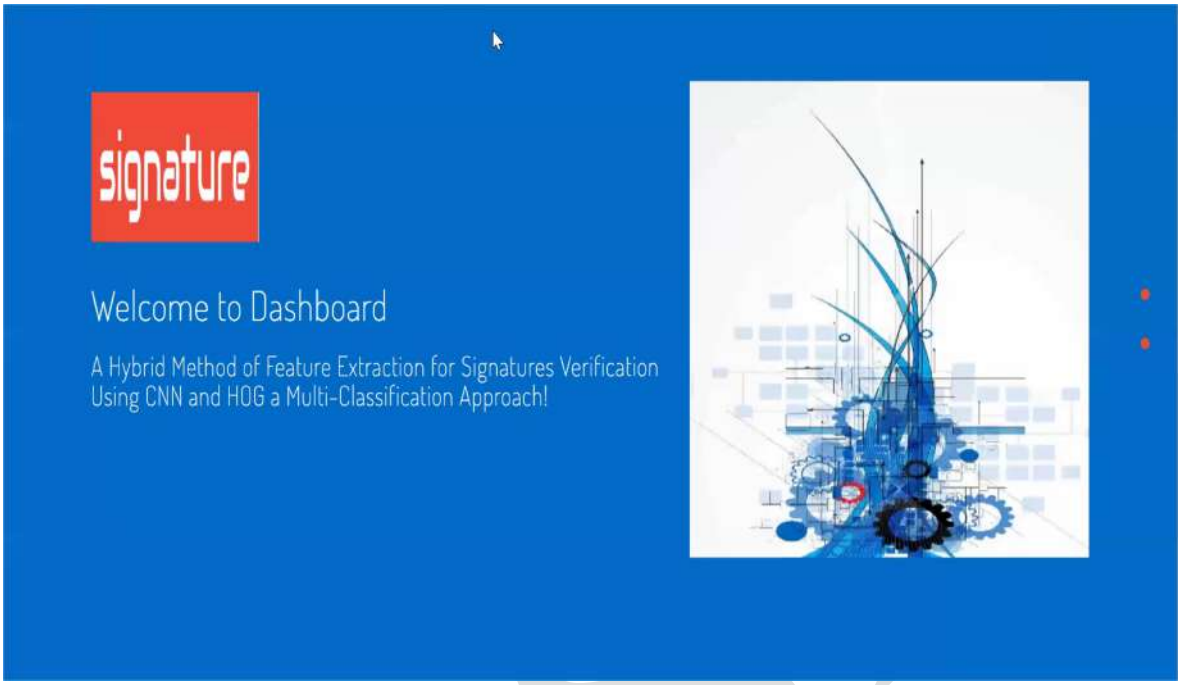

Sort
View
Name
Date modified
Type
Size
.ipynb_checkpoints 11-10-2023 19:44 File folder
archive 08-10-2023 20:11 File folder
sample 11-10-2023 19:58 File folder
static 11-10-2023 19:54 File folder
templates 13-10-2023 21:10 File folder
UTSig_Crop 10-10-2023 12:43 File folder
accuracy 09-10-2023 15:55 PNG File 446 KB
accuracy1 11-10-2023 04:01 PNG File 478 KB
app 11-10-2023 19:51 Python Source File 6 KB
CEDAR 11-10-2023 10:06 Jupyter Source File 479 KB
cnn.h5 09-10-2023 11:45 H5 File 1,67,959 KB
cnn_ut.h5 10-10-2023 15:22 H5 File 1,11,987 KB
loss 09-10-2023 15:55 PNG File 295 KB
loss1 11-10-2023 04:01 PNG File 295 KB
model.pkl 09-10-2023 16:18 PKL File 92 KB

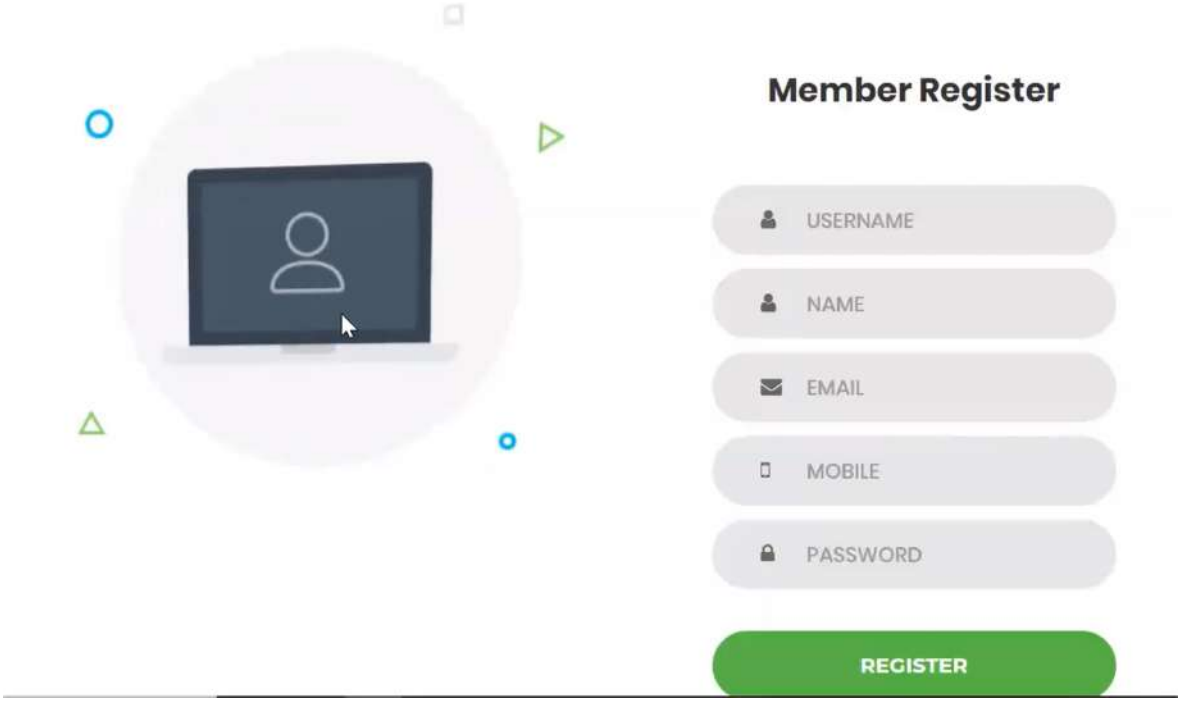

Extension
Copy Address
Copy Address as Text
Edit Address
Delete History
View
Name
Type
Size
.ipynb_checkpoints File folder
archive 08-10-2023 20:11 File folder
sample 11-10-2023 19:58 File folder
static 11-10-2023 19:54 File folder
templates 13-10-2023 21:10 File folder
UTSig_Crop 10-10-2023 12:43 File folder
accuracy 09-10-2023 15:55 PNG File 446 KB
accuracy1 11-10-2023 04:01 PNG File 478 KB
app 11-10-2023 19:51 Python Source File 6 KB
CEDAR 11-10-2023 10:06 Jupyter Source File 479 KB
cnn.h5 09-10-2023 11:45 H5 File 1,67,959 KB

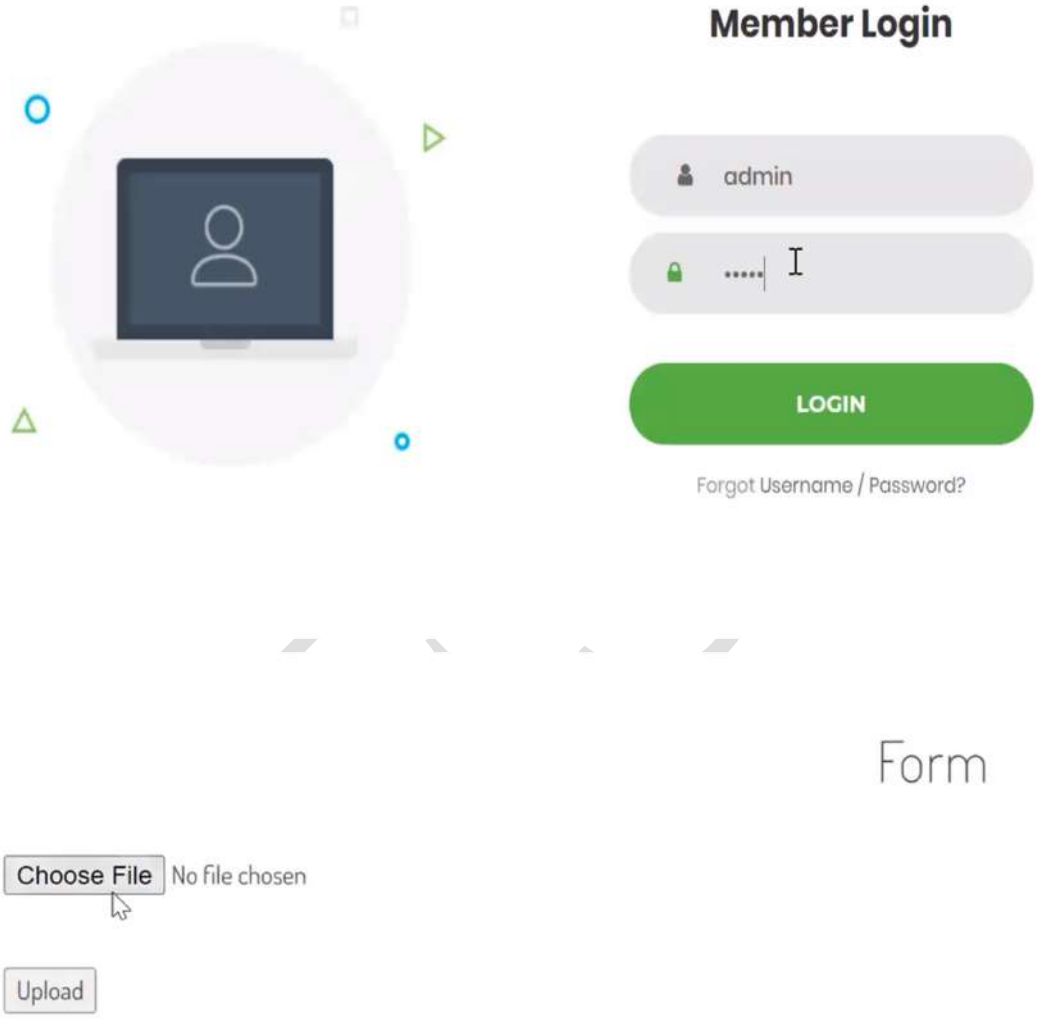

Select Anaconda Prompt (Anaconda3) - python app.py
(base) C:\Users\TruProjects>cd C:\Users\TruProjects\Desktop\02102023\17-A Hybrid Method of Feature Extraction for Signat
ures Verification Using CNN and HOG a Multi-Classification Approach\Extension
(base) C:\Users\TruProjects\Desktop\02102023\17-A Hybrid Method of Feature Extraction for Signatures Verification Using
CNN and HOG a Multi-Classification Approach\Extension>python app.py
2024-01-31 16:00:09.179157: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic libra
ry 'cudart64_110.dll'; dlerror: cudart64_110.dll not found
2024-01-31 16:00:09.179904: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dlerror if you do n
ot have a GPU set up on your machine.

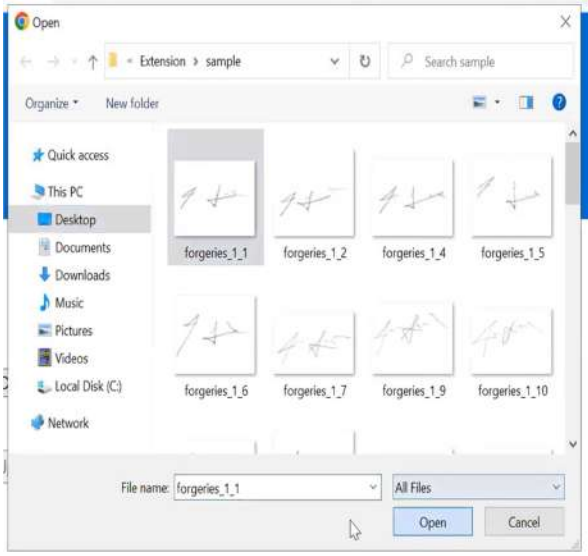

To enable them in other operations, rebuild Tensorflow with the appropriate compiler flags.
* Serving Flask app "app" (lazy loading)
* Environment: production
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)


http://127.0.0.1:5000/
Home - http://127.0.0.1:5000
http://127.0.0.1:5000/ - Google Search
Login V1 - http://127.0.0.1:5000/login
Result - http://127.0.0.1:5000/predict
Form - http://127.0.0.1:5000/signin?user=admin&password=admin
Google
Search Google or type a URL
Add shortcut


signature
Welcome to Dashboard
A Hybrid Method of Feature Extraction for Signatures Verification Using CNN and HOG a Multi-Classification Approach!


Member Register
USERNAME
NAME
EMAIL
MOBILE
PASSWORD
REGISTER


Member Login
admin
LOGIN
Forgot Username / Password?
Form
Choose File
No file chosen
Upload



The Predicted as :

Forgery

The Predicted as :

Geniune

CONCLUSION

The paper's conclusion asserts that the suggested hybrid approach for extracting features in offline signature verification systems includes CNN and HOG approaches, and is then followed by a feature selection algorithm.Three classifiers, namely LSTM, SVM, and KNN, were used for evaluation. Our suggested model demonstrated excellent accuracy when tested with the UTSig dataset and the CEDAR dataset. The findings indicated that the model performed well in terms of both performance and predictive capacity. It achieved a high level of accuracy in distinguishing between authentic and forged signatures, even for skilled forgeries. The research emphasizes the importance of the feature extraction step in offline signature verification systems and proposes that enhancing the feature extraction process might enhance the performance and predictive capabilities of these systems in the future.

REFERENCES

[1] F. M. Alsuhimat and F. S. Mohamad, ‘‘Offline signature verification using long short-term memory and histogram orientation gradient,’’ Bull. Elect. Eng. Inform., vol. 12, no. 1, pp. 283–292, 2023.

[2] M. Ajij, S. Pratihar, S. R. Nayak, T. Hanne, and D. S. Roy, ‘‘Off-line signature verification using elementary combinations of directional codes from boundary pixels,’’ Neural Comput. Appl., vol. 35, pp. 4939–4956, Mar. 2021, doi: 10.1007/s00521-021-05854-6.

[3] A. Q. Ansari, M. Hanmandlu, J. Kour, and A. K. Singh, ‘‘Online signature verification using segment-level fuzzy modelling,’’ IET Biometrics, vol. 3, no. 3, pp. 113–127, 2014.

[4] K. Cpałka and M. Zalasiński, ‘‘On-line signature verification using vertical signature partitioning,’’ Expert Syst. Appl., vol. 41, no. 9, pp. 4170–4180, 2014.

[5] K. Cpałka, M. Zalasiński, and L. Rutkowski, ‘‘A new algorithm for identity verification based on the analysis of a handwritten dynamic signature,’’ Appl. Soft Comput., vol. 43, no. 1, pp. 47–56, Jun. 2016.

[6] E. Griechisch, M. I. Malik, and M. Liwicki, ‘‘Online signature verification based on Kolmogorov–Smirnov distribution distance,’’ in Proc. 14th Int. Conf. Frontiers Handwriting Recognit., Sep. 2014, pp. 738–742.

[7] N. Sae-Bae and N. Memon, ‘‘Online signature verification on mobile devices,’’ IEEE Trans. Inf. Forensics Security, vol. 9, no. 6, pp. 933–947, Jun. 2014.

[8] S. Chen and S. Srihari, ‘‘A new off-line signature verification method based on graph matching,’’ in Proc. Int. Conf. Pattern Recognit. (ICPR), 2006, pp. 869–872.

[9] M. A. Ferrer, J. B. Alonso, and C. M. Travieso, ‘‘Offline geometric parameters for automatic signature verification using fixed-point arithmetic,’’ IEEE Trans. Pattern Anal. Mach. Intell., vol. 27, no. 6, pp. 993–997, Jun. 2005.

[10] Y. Guerbai, Y. Chibani, and B. Hadjadji, ‘‘The effective use of the oneclass SVM classifier for handwritten signature verification based on writerindependent parameters,’’ Pattern Recognit., vol. 48, no. 1, pp. 103–113, 2015.

[11] R. Larkins and M. Mayo, ‘‘Adaptive feature thresholding for off-line signature verification,’’ in Proc. 23rd Int. Conf. Image Vis. Comput. New Zealand, Nov. 2008, pp. 1–6.

[12] H. Lv, W. Wang, C. Wang, and Q. Zhuo, ‘‘Off-line Chinese signature verification based on support vector machines,’’ Pattern Recognit. Lett., vol. 26, no. 15, pp. 2390–2399, Nov. 2005.

[13] Y. Serdouk, H. Nemmour, and Y. Chibani, ‘‘New off-line handwritten signature verification method based on artificial immune recognition system,’’ Expert Syst. Appl., vol. 51, pp. 186–194, Jun. 2016.

[14] F. E. Batool, M. Attique, M. Sharif, K. Javed, M. Nazir, A. A. Abbasi, Z. Iqbal, and N. Riaz, ‘‘Offline signature verification system: A novel technique of fusion of GLCM and geometric features using SVM,’’ Multimedia Tools Appl., pp. 1–20, Apr. 2020, doi: 10.1007/s11042-020-08851-4.

[15] F. M. Alsuhimat and F. S. Mohamad, ‘‘Histogram orientation gradient for offline signature verification via multiple classifiers,’’ Nveo-Natural Volatiles Essential OILS J., vol. 8, no. 6, pp. 3895–3903, 2021.

[16] N. M. Tahir, N. Adam, U. I. Bature, K. A. Abubakar, and I. Gambo, ‘‘Offline handwritten signature verification system: Artificial neural network approach,’’ Int. J. Intell. Syst. Appl., vol. 1, no. 1, pp. 45–57, 2021.

[17] A. B. Jagtap, D. D. Sawat, R. S. Hegadi, and R. S. Hegadi, ‘‘Verification of genuine and forged offline signatures using Siamese neural network (SNN),’’ Multimedia Tools Appl., vol. 79, nos. 47–48, pp. 35109–35123, Dec. 2020.

[18] B. Kiran, S. Naz, and A. Rehman, ‘‘Biometric signature authentication using machine learning techniques: Current trends, challenges and opportunities,’’ Multimedia Tools Appl., vol. 79, no. 1, pp. 289–340, 2020.

[19] M. Sharif, M. A. Khan, M. Faisal, M. Yasmin, and S. L. Fernandes, ‘‘A framework for offline signature verification system: Best features selection approach,’’ Pattern Recognit. Lett., vol. 139, pp. 50–59, Nov. 2020.

[20] N. Sharma, S. Gupta, and P. Metha, ‘‘A comprehensive study on offline signature verification,’’ in Proc. J. Phys., Conf., 2021, Art. no. 012044, doi: 10.1088/1742-6596/1969/1/012044.

[21] H. H. Kao and C. Y. Wen, ‘‘An offline signature verification and forgery detection method based on a single known sample and an explainable deep learning approach,’’ Appl. Sci., vol. 10, no. 1, p. 3716, 2020.

[22] M. K. Kalera, S. Srihari, and A. Xu, ‘‘Offline signature verification and identification using distance statistics,’’ Int. J. Pattern Recognit. Artif. Intell., vol. 18, no. 7, pp. 1339–1360, 2004.

[23] B. Kovari and H. Charaf, ‘‘A study on the consistency and significance of local features in off-line signature verification,’’ Pattern Recognit. Lett., vol. 34, no. 3, pp. 247–255, 2013.

[24] T.-A. Pham, H.-H. Le, and N.-T. Do, ‘‘Offline handwritten signature verification using local and global features,’’ Ann. Math. Artif. Intell., vol. 75, nos. 1–2, pp. 231–247, Oct. 2015.

[25] Z. ZulNarnain, M. S. Rahim, N. F. Ismail, and M. M. Arsad, ‘‘Triangular geometric feature for offline signature verification,’’ Int. J. Comput. Inf. Eng., vol. 10, no. 3, pp. 485–488, 2016.

[26] R. K. Bharathi and B. H. Shekar, ‘‘Off-line signature verification based on chain code histogram and support vector machine,’’ in Proc. Int. Conf. Adv. Comput., Commun. Informat. (ICACCI), Aug. 2013, pp. 2063–2068.

[27] V. Nguyen, Y. Kawazoe, T. Wakabayashi, U. Pal, and M. Blumenstein, ‘‘Performance analysis of the gradient feature and the modified direction feature for off-line signature verification,’’ in Proc. 12th Int. Conf. Frontiers Handwriting Recognit., Nov. 2010, pp. 303–307.

[28] R. Kumar, J. D. Sharma, and B. Chanda, ‘‘Writer-independent off-line signature verification using surroundedness feature,’’ Pattern Recognit. Lett., vol. 33, no. 3, pp. 301–308, Feb. 2012.

[29] M. Hanmandlu, M. H. M. Yusof, and V. K. Madasu, ‘‘Off-line signature verification and forgery detection using fuzzy modeling,’’ Pattern Recognit., vol. 38, no. 3, pp. 341–356, 2005.

[30] N. Jiang, J. Xu, W. Yu, and S. Goto, ‘‘Gradient local binary patterns for human detection,’’ in Proc. IEEE Int. Symp. Circuits Syst. (ISCAS), May 2013, pp. 978–981.