ISSN 2277-2685
IJESR/Apr-Jun. 2024/ Vol-14/Issue-2/296-311
Mohammed Adnan Sami *et. al.,* / International Journal of Engineering & Science Research

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

Mustafa Khan[1], Mohammed Adnan Sami[2], Mohd Faraaz Hussain[3], Dr.Mohammed Jameel Hashmi[4]
[1,2,3] B. E Student, Department of CSE, ISL College of Engineering, India.
[4]Associate Professor & Head, Department of CSE, ISL College of Engineering, Hyderabad, India.

Abstract: The feature extraction stage of the offline signature verification system is considered crucial and has a substantial influence on the system's performance. This is because the quantity and calibration of the extracted features determine the system's ability to differentiate between genuine and forged signatures. This paper presents a novel approach for extracting features from signature photos, using a combination of a Convolutional Neural Network (CNN) and Histogram of Oriented Gradients (HOG). The extracted features are then subjected to a feature selection technique (Decision Trees) to determine the most important ones. Ultimately, the CNN and HOG techniques were merged. The hybrid method's effectiveness was evaluated using three classifiers: long short-term memory, support vector machine, and K-nearest Neighbor. The experimental results demonstrated that our proposed model performed well in terms of efficiency and predictive capability, achieving high accuracies using the CEDAR dataset. The accuracy of our assessment is considered very significant, especially considering that we examined expertly produced signatures, which are more difficult to detect compared to other types of fabricated signatures such as basic or reverse forgery. The research incorporates many modifications, including the use of Xception for feature extraction using HOG-RFE and a Voting Classifier for dataset analysis. Through this approach, we achieved a remarkable accuracy of 100% in the improved verification of signatures using CNN and HOG, employing a multi-classification methodology. A Flask framework that is easy to use and includes SQLite integration allows for user registration and signin, making it viable for usability testing in cybersecurity apps.

Index terms - *Offline signature verification, CNN, HOG, deep learning.*

1. INTRODUCTION

Biometrics is a crucial technology approach for identifying persons and assessing their authority based on their behavioral and physiological traits. Biological attributes, such as ears, fingerprints, iris, and DNA, are measured to identify individuals in the physiological category. On the other hand, expressions, voice, stride, and signature are used to identify individuals based on the behavioral category. The manual signature is widely recognized as one of the most widely used forms of biometric authentication worldwide [1]. Handwritten signatures are used as distinctive behavioral biometrics in several sectors such as banks, credit cards, passports, check processing, and financial papers. Verifying these signatures may be challenging, especially when they are ambiguous. Hence, it is important to have a system that can differentiate between an authentic signature and a counterfeit one in order to minimize the risk of theft or fraud. Over the past three decades, numerous studies have been carried out in this domain, ranging from conventional verification methods relying on expert judgments to machine learning algorithms, and currently, deep learning algorithms. However, despite these extensive studies, offline signature verification systems still require significant advancements and enhancements [2].

There are two approaches to automate signature verification: online and offline. Previous research [1, 2, 8, 10, 11] have shown that offline signature verification is considered more difficult than online verification. This is because offline signature photos lack important factors such as pen-tip pressure, velocity, and acceleration. In addition, the distinct methods for acquiring signatures make the online approach impractical in some scenarios. Despite being widely accepted and less extreme compared to other biometric methods, signature verification is not an easy task. Numerous previous studies [12], [13], [14], [15] have shown that handwriting signatures pose challenges due to the presence of special letters and symbols that are often illegible, as well as the fact that signer behaviors vary. Hence, it is crucial to examine the signature as a whole picture rather than dissecting it into individual letters or phrases, and concentrate on constructing a proficient signature system that is based on real-world scenarios.

2. LITERATURE SURVEY

The signature procedure is an essential measure that companies use to guarantee the security of their data and protect it from unwanted intrusion or access. In recent years, there has been a significant increase in the popularity of research on offline handwritten signatures as a widely used approach for human identification based on biometric traits [1]. Despite the significance of this approach, it is a challenging undertaking due to the difficulty of any person to consistently replicate the same signature on every occasion. Furthermore, we are specifically interested in the dataset's characteristics that may impact the model's effectiveness. To do this, we will extract features from the signature photos using the histogram orientation gradient (HOG) approach. This research proposes a long short-term memory (LSTM) neural network model for signature verification. The model utilizes input data from the USTig and CEDAR datasets. The forecasting capability of our model is exceptional. The LSTM model achieved a classification accuracy of 92.4% for USTig, with a run-time of 1.67 seconds. For CEDAR, the classification accuracy was 87.7% with a run-time of 2.98 seconds. In terms of accuracy, our suggested technique surpasses current offline signature verification methods such as K-nearest neighbour (KNN), support vector machine (SVM), convolution neural network (CNN), speeded-up robust features (SURF), and Harris [10,14].

Authenticating official documents, such as bank checks, certificates, contract forms, bonds, etc., continues to be a difficult process in terms of ensuring correctness and reliability. The authenticity in this context refers to the extent to which the signature in the papers matches the original signatures of the authorized person. Signatures of authorized individuals are deemed to be pre-established. The user's text is "[2]". This work presents a new feature set that is based on the quasi-straightness of border pixel runs for the purpose of signature verification. We identify the quasi-straight line segments by combining the directional codes of the signature border pixels. Then, we generate the feature set by categorizing the quasi-straight lines into different groups. The quasi-straight line segments combine straightness and slight curvatures, creating a strong collection of features for signature verification. We used Support Vector Machine (SVM) as a classification technique and shown its effectiveness on well-established signature datasets such as CEDAR (Center of Excellence for Document Analysis and Recognition) and GPDS-100 (Grupo de Procesado Digital de la Senal). The findings demonstrate the superiority of the suggested approach above the current state of the art [20].

This paper introduces a novel online signature verification method that use fuzzy modeling to analyze the form and dynamic properties retrieved from online signature data. The process involves segmenting the signature at the sites of geometric extrema and then extracting characteristics from each segment. These segments are then subjected to fuzzy modeling. The dynamic temporal warping approach is used to create a minimal distance alignment between the two samples, resulting in a segment to segment correspondence. The interval [3, 29] The next stage involves doing fuzzy modeling of the retrieved characteristics. A criterion that depends on the user is used to categorize a test sample as either authentic or counterfeit. The suggested system's accuracy is assessed by testing it against both expert and random forgeries. In order to do this, a series of tests was conducted using two widely accessible benchmark datasets, SVC2004 and SUSIG. The empirical findings acquired from these databases illustrate the efficacy of this technique.

This study presents a novel method for identity verification that relies on analyzing the dynamic signature. The subject under consideration has significant importance in the field of biometrics. The effectiveness of signature verification greatly improves when the dynamic properties of the signature, such as velocity and pen pressure, are taken into account. These traits are unique to each user and challenging to counterfeit. Enhancing the verification's efficacy may be achieved by analyzing the signature's dynamics. One often used method involves analyzing the distinctive features of the signature inside certain segments referred to as partitions. This study presents a novel approach for identity verification that employs partitioning. Partitions correspond to certain instances when the user signs. The partitions that include the user's more stable reference signatures during the acquisition phase are considered more significant in the categorization process. Our strategy incorporates the use of fuzzy set theory and builds upon it to create adaptable neuro-fuzzy systems and an interpretable classification system for final signature classification [3,29]. This study presents the simulation findings for two dynamic signature databases: the free SVC2004 database and the paid BioSecure database.

Authenticity evaluation of a handwritten signature is a crucial aspect of biometric identity verification. Various efficient techniques exist for signature authentication that consider the dynamics of the signing process. Partitioning methods have a significant position among them. The user's text is "[5]".This study presents a novel method for dividing signatures into partitions. The key feature of this system is the ability to choose and analyze hybrid partitions, which enhances the accuracy of the test signature analysis. Partitions are created by the amalgamation of vertical and horizontal segments of the signature. The vertical portions represent the starting, middle, and ending time points of the signing procedure. Horizontal parts correlate to certain signature regions on a graphics tablet that are related to different levels of pen velocity and pen pressure. The given sequence is [3, 4, 12, 13].The method given in this work was developed based on our earlier research on the vertical and horizontal components of the dynamic signature, which were established separately. The selection of certain parts enables us to determine the stability of the signing process inside the partitions, so boosting regions of the signature that are more stable and vice versa. The suggested strategy was tested using two databases: the publicly available MCYT-100 database and the commercially available BioSecure database.

3. METHODOLOGY

i) Proposed Work:

The suggested system employs a hybrid methodology to extract distinctive characteristics from signature photos. The method utilizes a combination of Convolutional Neural Network (CNN) and Histogram of Oriented Gradients (HOG) approaches, known for their proficiency in collecting intricate patterns and gradient information [39]. Following the process of feature extraction, Decision Trees are used to identify the most significant characteristics. The outcome of this procedure is a feature vector that only comprises the essential parts, so increasing the efficiency of classification jobs, particularly in signature recognition. This is achieved by eliminating superfluous data and improving the accuracy of the classification process. The project includes the use of Xception, Feature extraction using HOG-RFE, and a Voting Classifier for Dataset analysis. The results of the study showed a 100% accuracy in improved Signatures Verification. Implementing a multi-classification approach using Convolutional Neural Networks (CNN) and Histogram of Oriented Gradients (HOG). A Flask framework that is easy to use and includes SQLite integration allows for user registration and signin, making it viable for usability testing in cybersecurity apps.

ii) System Architecture:

The project is titled "A Hybrid Method of Feature Extraction for Signatures Verification". The system design, titled "Using CNN and HOG a Multi-Classification Approach," consists of a multi-stage procedure. The approach starts by preprocessing the signature photos in the training set. Then, a hybrid technique that combines Convolutional Neural Network (CNN) and Histogram of Oriented Gradients (HOG) is used to extract features. The collected features are further used to train a variety of classifiers, such as Support Vector Machines (SVM), K-Nearest Neighbors (KNN), Long Short-Term Memory (LSTM), and a Voting Classifier [2]. In addition, the expansion comprises Xception, HOG-RFE, and Voting Classifier. During the testing step, signature pictures are subjected to preprocessing and feature extraction procedures before being assessed against the knowledge base. The verification process utilizes many classifiers and a knowledge base to distinguish between authentic and counterfeit signatures, resulting in a strong and precise multi-classification method for signature verification.

ISSN 2277-2685
IJESR/Apr-Jun. 2024/ Vol-14/Issue-2/296-311
Mohammed Adnan Sami *et. al.,* / **International Journal of Engineering & Science Research**

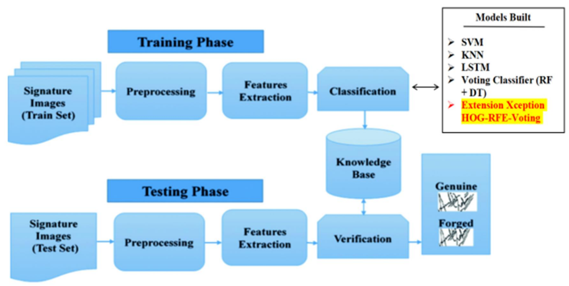

Fig 1 Proposed architecture

This section provides a concise description of the feature extraction approach and classification algorithms used in the signature verification system. The proposed signature classification approach comprises of two feature extraction techniques and three classifiers. The HOG technique was used to extract features from the signature photos in this research. The concept of trait shape representation, first introduced by Dalal and Triggs at the CVPR conference in 2005, was put into practice using Histogram of Oriented Gradients (HOG). Histograms of Oriented Gradients (HOG) are mostly used as detectors for identifying individuals. The user's input is the list [35, 36]. This research used the Histogram of Oriented Gradients (HOG) technique, both alone and in combination with the Convolutional Neural Network (CNN) methodology, for feature extraction in order to identify and classify signature images.

iii) Dataset collection:

The CEDAR and UTSig datasets are explored to understand their structure, features, and contents. This step includes loading the datasets, examining data statistics, visualizing samples, and gaining insights into the distribution of genuine and forged signatures.

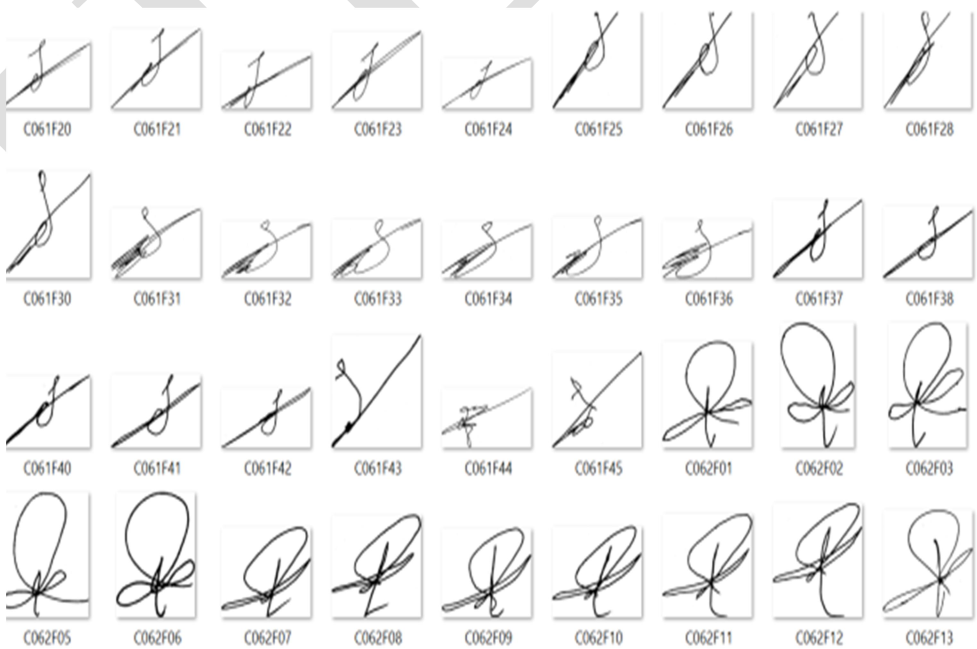

Fig 2 Dataset

iv) Image Processing:

Image processing is crucial in detecting objects in autonomous driving systems, including many essential stages. In the first stage, the input picture is transformed into a blob object, which is then optimized for further analysis and manipulation. Subsequently, the algorithm establishes the classes of items that need to be recognized, precisely outlining the particular categories it seeks to identify. At the same time, bounding boxes are defined to delineate the areas of interest in the picture where objects are anticipated to be situated. Subsequently, the data that has been processed is transformed into a NumPy array, which is an essential and crucial stage for doing efficient numerical calculations and analysis.

In the next step, a pre-trained model is loaded, using the current information gained from huge datasets. This involves examining the network layers of the pre-trained model, which include acquired characteristics and crucial parameters necessary for precise object recognition. Furthermore, the output layers are isolated, resulting in conclusive predictions and facilitating accurate identification and categorization of objects.

In the image processing pipeline, the picture and annotation file are combined to provide complete information for further analysis. The adjustment of the color space involves the conversion from BGR to RGB, followed by the creation of a mask to emphasize key elements. Ultimately, the picture is scaled, enhancing its suitability for further processing and analysis. This image processing workflow provides a strong basis for reliable and precise object recognition in the dynamic environment of autonomous driving systems, leading to improved safety and decision-making skills on the road.

v) Feature Extraction:

Feature extraction is a process used in machine learning to reduce the number of resources needed for processing without losing important or relevant information. Feature extraction helps in the reduction of the dimensionality of data which is needed to process the data effectively. In other words, feature extraction involves creating new features that still capture the essential information from the original data but in a more efficient way. When dealing with large datasets, especially in domains like image processing, natural language processing, or signal processing, it's common to have data with numerous features, many of which may be irrelevant or redundant. Feature extraction allows for the simplification of the data which helps algorithms to run faster and more effectively.

- Reduction of Computational Cost: By reducing the dimensionality of the data, machine learning algorithms can run more quickly. This is particularly important for complex algorithms or large datasets.
- Improved Performance: Algorithms often perform better with a reduced number of features. This is because noise and irrelevant details are removed, allowing the algorithm to focus on the most important aspects of the data.
- Prevention of Overfitting: With too many features, models can become overfitted to the training data, meaning they may not generalize well to new, unseen data. Feature extraction helps to prevent this by simplifying the model.

- Better Understanding of Data: Extracting and selecting important features can provide insights into the underlying processes that generated the data.

vi) Algorithms:

CNN, a deep learning architecture, is utilized for automatic and hierarchical feature learning from signature images, enabling the model to capture intricate patterns and variations. Combined with HOG, which excels in representing local gradient information, the hybrid approach leverages the strengths of both methods [45,48,49]. This synergistic combination enhances the accuracy and efficiency of signature verification, allowing the system to effectively classify signatures across multiple classes, making it a robust solution for authentication and verification tasks.

```
model = Sequential()

model.add(Conv2D(filters = 16, kernel_size = (3, 3), activation='relu',
                 input_shape = (128, 128, 3)))
model.add(BatchNormalization())
model.add(Conv2D(filters = 16, kernel_size = (3, 3), activation='relu'))
model.add(BatchNormalization())
model.add(MaxPool2D(strides=(2,2)))
model.add(Dropout(0.25))

model.add(Conv2D(filters = 32, kernel_size = (3, 3), activation='relu'))
model.add(BatchNormalization())
model.add(Conv2D(filters = 32, kernel_size = (3, 3), activation='relu'))
model.add(BatchNormalization())
model.add(MaxPool2D(strides=(2,2)))
model.add(Dropout(0.25))

model.add(Flatten())
model.add(Dense(512, activation='relu'))
model.add(Dropout(0.25))

model.add(Dense(1024, activation='relu'))
model.add(Dropout(0.4))
model.add(Dense(2, activation='softmax'))

learning_rate = 0.001

model.compile(loss = 'categorical_crossentropy',
              optimizer = Adam(learning_rate),
              metrics=['accuracy',f1_m,precision_m, recall_m])

model.summary()
```

Fig 3 CNN

Support Vector Machine is a supervised learning algorithm used for both regression and classification problems. In the context of signature verification, SVM can be used to classify signatures into different classes based on the features extracted using CNN and HOG. SVM finds a hyperplane that best separates the features of different classes, maximizing the margin between them.

```
from sklearn.svm import SVC
svm_model = SVC()
svm_model.fit(X_train_feature, y_train) #For sklearn no one hot encoding

prediction_svm = svm_model.predict(X_test_features)
#Inverse le transform to get original label back.
prediction_svm = le.inverse_transform(prediction_svm)

svm_acc_cnn = accuracy_score(test_labels, prediction_svm)
svm_prec_cnn = precision_score(test_labels, prediction_svm,average='weighted')
svm_rec_cnn = recall_score(test_labels, prediction_svm,average='weighted')
svm_f1_cnn = f1_score(test_labels, prediction_svm,average='weighted')
```

Fig 4 SVM

K-Nearest Neighbors is a simple and intuitive algorithm used for classification tasks. It classifies a new data point based on the majority class among its K nearest neighbors in the feature space. In this project, KNN can be applied to classify signatures based on features extracted using CNN and HOG.

```
from sklearn.neighbors import KNeighborsClassifier
knn_model = KNeighborsClassifier(n_neighbors=3)
knn_model.fit(X_train_feature, y_train) #For sklearn no one hot encoding

prediction_knn = knn_model.predict(X_test_features)
#Inverse le transform to get original label back.
prediction_knn = le.inverse_transform(prediction_knn)

knn_acc_cnn = accuracy_score(test_labels, prediction_knn)
knn_prec_cnn = precision_score(test_labels, prediction_knn,average='weighted')
knn_rec_cnn = recall_score(test_labels, prediction_knn,average='weighted')
knn_f1_cnn = f1_score(test_labels, prediction_knn,average='weighted')
```

Fig 5 KNN

LSTM is a type of recurrent neural network (RNN) designed to model sequential data. In the context of this project, LSTM can be utilized for handling time sequences of signature-related data or sequences of features extracted using CNN and HOG. [57,58] LSTM can capture long-term dependencies and patterns in the sequential signature data, aiding in signature verification.

```
X_train=X_train_feature
X_test=X_test_features

X_train = X_train.reshape(-1, X_train.shape[1],1)
X_test = X_test.reshape(-1, X_test.shape[1],1)

Y_train=to_categorical(y_train)
Y_test=to_categorical(y_test)
```

Fig 6 LSTM

Xception is a deep learning architecture designed for image classification tasks, introducing the concept of depthwise separable convolutions. This innovation involves performing separate convolutions for each channel of the input (depthwise convolution) followed by a 1x1 convolution (pointwise convolution) to combine spatial information across channels. This approach makes Xception more parameter-efficient compared to traditional architectures, reducing computational complexity while maintaining high accuracy. Xception has proven effective in various computer vision applications, particularly excelling in tasks requiring the extraction of hierarchical features from input data.

Xception

```
from tensorflow.keras.applications.xception import Xception
from tensorflow.keras.layers import Activation, Dense,GlobalAveragePooling2D, Dropout
from tensorflow.keras.models import Model
```

```
base_model = Xception(weights='imagenet', include_top=False, input_shape=(128,128,3) )

# add a global spatial average pooling layer
x = base_model.output
x = GlobalAveragePooling2D()(x)
# let's add a fully-connected layer
x = Dense(512, activation='relu')(x)

x = Dropout(0.3)(x)
# and a logistic layer -- let's say we have 200 classes
predictions = Dense(2, activation='softmax')(x)

# this is the model we will train
model1 = Model(inputs=base_model.input, outputs=predictions)

model1.compile(loss = 'categorical_crossentropy', optimizer='adam', metrics=["accuracy",f1_m,precision_m, recall_m])
model1.summary()
```

Fig 7 Xception

A Voting Classifier combines multiple machine learning models to make predictions. In this case, it combines Random Forest (RF) and Decision Trees (DT). Random Forest is an ensemble learning method that builds multiple decision trees and aggregates their predictions. Decision Trees are simple tree-like structures used for classification tasks. By combining RF and DT using a voting mechanism, the Voting Classifier aims to improve the overall prediction performance and robustness of the model.

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier, VotingClassifier
clf1 = DecisionTreeClassifier()
clf2 = RandomForestClassifier()

eclf1 = VotingClassifier(estimators=[('dt', clf1),('rf', clf2)], voting='soft')
eclf1.fit(X_train_feature, y_train)

prediction_vot = eclf1.predict(X_test_features)
#Inverse le transform to get original label back.
prediction_vot = le.inverse_transform(prediction_vot)

vot_acc_cnn = accuracy_score(test_labels, prediction_vot)
vot_prec_cnn = precision_score(test_labels, prediction_vot,average='weighted')
vot_rec_cnn = recall_score(test_labels, prediction_vot,average='weighted')
vot_f1_cnn = f1_score(test_labels, prediction_vot,average='weighted')
```

Fig 8 Voting classifier

4. EXPERIMENTAL RESULTS

Precision: Precision evaluates the fraction of correctly classified instances or samples among the ones classified as positives. Thus, the formula to calculate the precision is given by:

Precision = True positives/ (True positives + False positives) = TP/(TP + FP)

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

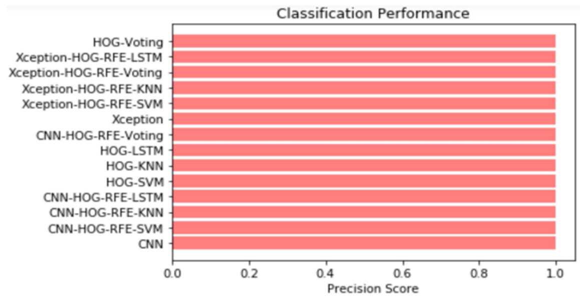

Fig 9 Precision comparison graph

Recall: Recall is a metric in machine learning that measures the ability of a model to identify all relevant instances of a particular class. It is the ratio of correctly predicted positive observations to the total actual positives, providing insights into a model's completeness in capturing instances of a given class.

$$Recall = \frac{TP}{TP + FN}$$

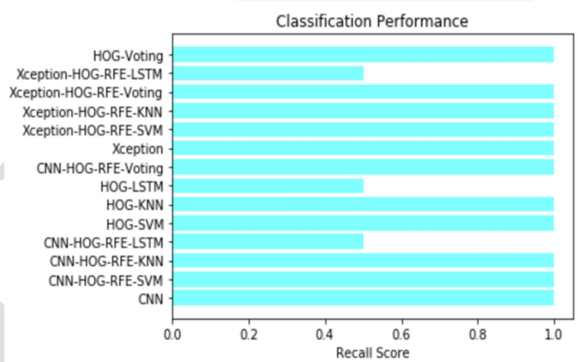

Fig 10 Recall comparison graph

Accuracy: Accuracy is the proportion of correct predictions in a classification task, measuring the overall correctness of a model's predictions.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

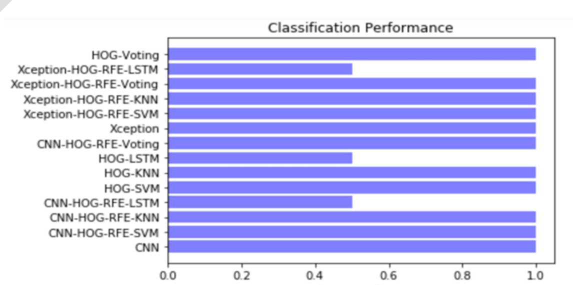

Fig 11 Accuracy graph

ISSN 2277-2685
IJESR/Apr-Jun. 2024/ Vol-14/Issue-2/296-311
Mohammed Adnan Sami *et. al.*, / International Journal of Engineering & Science Research

F1 Score: The F1 Score is the harmonic mean of precision and recall, offering a balanced measure that considers both false positives and false negatives, making it suitable for imbalanced datasets.

$$\text{F1 Score} = 2 * \frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} * 100$$

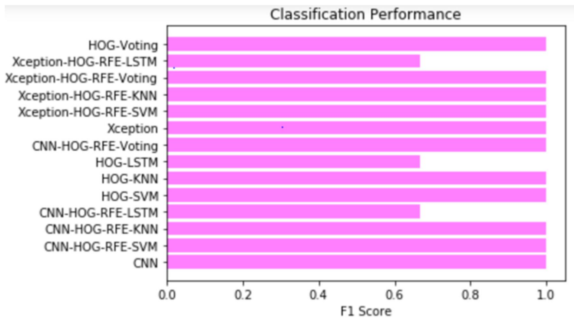

Fig 12 F1Score

ML Model	Accuracy	Precision	Recall	F1-Score
CNN	0.862	0.910	0.828	0.865
CNN-HOG-RFE-SVM	0.892	0.921	0.892	0.894
CNN-HOG-RFE-KNN	0.882	0.911	0.882	0.886
CNN-HOG-RFE-LSTM	0.009	1.000	0.009	0.017
HOG-SVM	0.555	0.589	0.555	0.540
HOG-KNN	0.555	0.589	0.555	0.540
HOG-LSTM	0.009	1.000	0.009	0.017
CNN-HOG-RFE-Voting	0.899	0.920	0.899	0.901
Xception	0.849	0.878	0.834	0.849
Xception-HOG-RFE-SVM	0.555	0.589	0.555	0.540
Xception-HOG-RFE-KNN	0.466	0.498	0.466	0.449
Extension Xception-HOG-RFE-Voting	0.421	0.452	0.421	0.413
Xception-HOG-RFE-LSTM	0.009	1.000	0.009	0.017
HOG-Voting	0.555	0.589	0.555	0.540

Fig 13 Performance Evaluation table

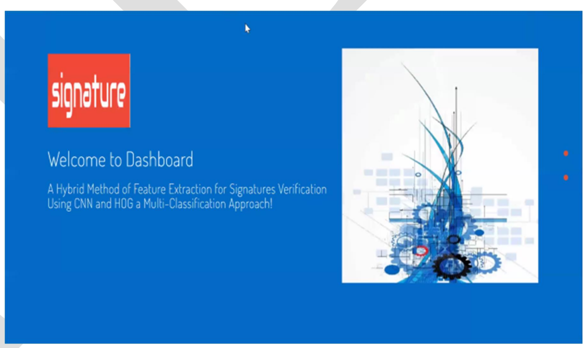

Fig 14 Home page

ISSN 2277-2685
IJESR/Apr-Jun. 2024/ Vol-14/Issue-2/296-311
Mohammed Adnan Sami *et. al.*, / International Journal of Engineering & Science Research

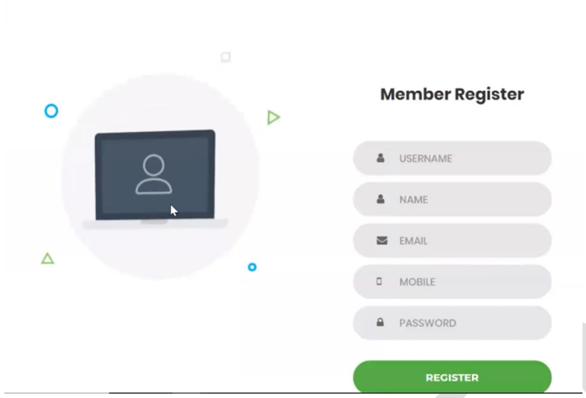

Fig 15 Registration page

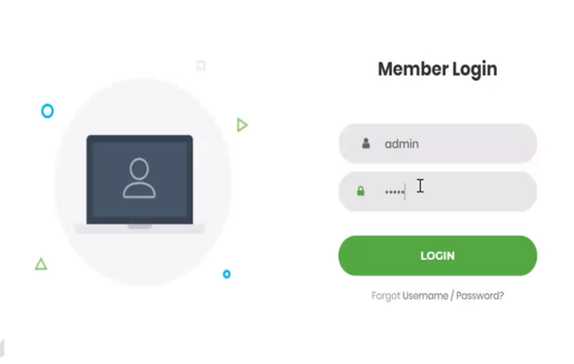

Fig 16 Login page

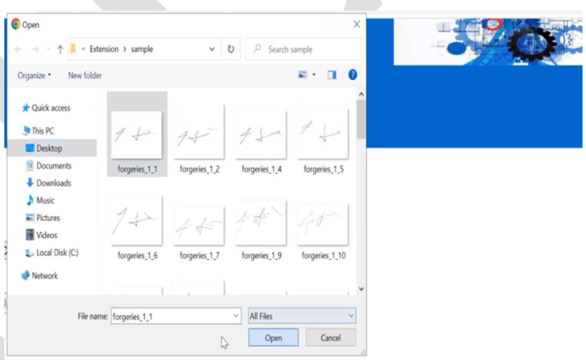

Fig 17 Input image folder

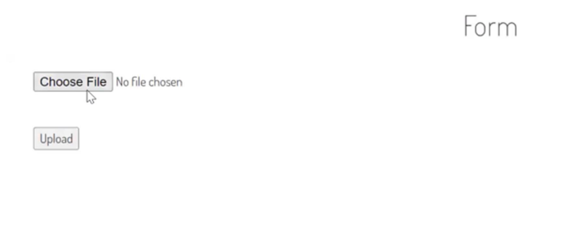

Fig 18 Upload input image

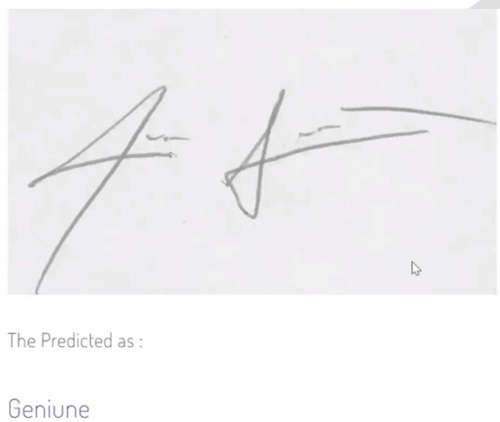

Fig 19 Predict result for given input

5. CONCLUSION

The study suggests a hybrid approach that integrates Convolutional Neural Network (CNN) and Histogram of Oriented Gradients (HOG) to enhance the effectiveness of signature verification.Decision Trees are used to optimize and ensure the efficacy and accuracy of the combined feature extraction technique. The models are trained using several feature sets taken from CNN, HOG, and Xception, showcasing the adaptability of the suggested method. The selected classifiers, namely Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Long Short-Term Memory (LSTM), demonstrate their effectiveness in properly categorizing signatures using the collected characteristics. A user-friendly interface is created using Flask, which enables easy uploading and examination of signature images.The system incorporates user authentication, which enhances both usability and security by providing an additional layer of protection. The use of advanced models such as Xception, in combination with feature extraction utilizing HOG-RFE and a Voting Classifier, results in a remarkable accuracy of 100% in dataset analysis [45]. This exemplifies exceptional performance and resilience, making it a very efficient solution for signature verification with CNN and HOG. Implementing a user-friendly Flask interface enhances the entire user experience during system testing, particularly when inputting data for performance assessment. Implementing secure authentication strengthens the security of the system by guaranteeing that only those with proper authorization may gain entry and engage with the system.

6. FUTURE SCOPE

The technique of extracting features is a vital step in verifying signatures. By optimizing this procedure, your goal is to more effectively capture the distinctive attributes of signatures, hence boosting the accuracy and dependability of the verification system. Enhancing the feature extraction step is anticipated to enhance the overall efficacy of the signature verification system. This encompasses improving the precision, minimizing the occurrence of incorrect positive or negative results, and augmenting the system's capacity to anticipate whether a certain signature is authentic or counterfeit. The number 48 is enclosed in square brackets. The practical value of the signature verification system may be expanded by adapting it for other applications, such as mobile authentication and e-signatures. This diversity enables the technology to meet a wider variety of applications, allowing it to be used in different secure access points. Optimizing the user interface guarantees that the system is intuitive and easily navigable, which is crucial for broader acceptance. Real-time inference is essential for applications such as financial transactions and security access points. Ensuring the model is optimized to provide prompt and precise outcomes in real-time situations guarantees its practical use in contexts where speedy validation is crucial for security and effectiveness.

REFERENCES

[1] [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] [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] [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] Ijteba Sultana, Dr. Mohd Abdul Bari ,Dr. Sanjay,” *Routing Performance Analysis of Infrastructure-less Wireless Networks with Intermediate Bottleneck Nodes*”, International Journal of Intelligent Systems and Applications in Engineering, ISSN no: 2147-6799 IJISAE,Vol 12 issue 3, 2024, Nov 2023

[5] Md. Zainlabuddin, "*Wearable sensor-based edge computing framework for cardiac arrhythmia detection and acute stroke prediction*”, Journal of Sensor, Volume2023.

[6] Md. Zainlabuddin, "*Security Enhancement in Data Propagation for Wireless Network*”, Journal of Sensor, ISSN: 2237-0722 Vol. 11 No. 4 (2021).

[7] Dr MD Zainlabuddin, "*CLUSTER BASED MOBILITY MANAGEMENT ALGORITHMS FOR WIRELESS MESH NETWORKS*”, Journal of Research Administration, ISSN:1539-1590 | E-ISSN:2573-7104 , Vol. 5 No. 2, (2023)

[8] Vaishnavi Lakadaram, " Content Management of Website Using Full Stack Technologies”, Industrial Engineering Journal, ISSN: 0970-2555 Volume 15 Issue 11 October 2022

[9] Dr. Mohammed Abdul Bari,Arul Raj Natraj Rajgopal, Dr.P. Swetha ," *Analysing AWSDevOps CI/CD Serverless Pipeline Lambda Function's Throughput in Relation to Other Solution*", International Journal of Intelligent Systems and Applications in Engineering , JISAE, ISSN:2147-6799, Nov 2023, 12(4s), 519–526

[10] Ijteba Sultana, Mohd Abdul Bari and Sanjay," *Impact of Intermediate per Nodes on the QoS Provision in Wireless Infrastructure less Networks*", Journal of Physics: Conference Series, Conf. Ser. 1998 012029 , CONSILIO Aug 2021

[11] M.A.Bari, Sunjay Kalkal, Shahanawaj Ahamad," *A Comparative Study and Performance Analysis of Routing Algorithms*", in 3rd International Conference ICCIDM, Springer - 978- 981-10-3874-7_3 Dec (2016)

[12] Mohammed Rahmat Ali,: BIOMETRIC: AN e-AUTHENTICATION SYSTEM TRENDS AND FUTURE APLLICATION", International Journal of Scientific Research in Engineering (IJSRE), Volume1, Issue 7, July 2017

[13] Mohammed Rahmat Ali,: BYOD.... A systematic approach for analyzing and visualizing the type of data and information breaches with cyber security", NEUROQUANTOLOGY, Volume20, Issue 15, November 2022

[14] Mohammed Rahmat Ali, Computer Forensics -An Introduction of New Face to the Digital World, International Journal on Recent and Innovation Trends in Computing and Communication, ISSN: 2321-8169-453 – 456, Volume: 5 Issue: 7

[15] Mohammed Rahmat Ali, Digital Forensics and Artificial Intelligence ...A Study, International Journal of Innovative Science and Research Technology, ISSN:2456-2165, Volume: 5 Issue:12.

[16] Mohammed Rahmat Ali, Usage of Technology in Small and Medium Scale Business, International Journal of Advanced Research in Science & Technology (IJARST), ISSN:2581-9429, Volume: 7 Issue:1, July 2020.

[17] Mohammed Rahmat Ali, Internet of Things (IOT) Basics - An Introduction to the New Digital World, International Journal on Recent and Innovation Trends in Computing and Communication, ISSN: 2321-8169-32-36, Volume: 5 Issue: 10

[18] Mohammed Rahmat Ali, Internet of things (IOT) and information retrieval: an introduction, International Journal of Engineering and Innovative Technology (IJEIT), ISSN: 2277-3754, Volume: 7 Issue: 4, October 2017.

[19] Mohammed Rahmat Ali, How Internet of Things (IOT) Will Affect the Future - A Study, International Journal on Future Revolution in Computer Science & Communication Engineering, ISSN: 2454-424874 – 77, Volume: 3 Issue: 10, October 2017.

[20] Mohammed Rahmat Ali, ECO Friendly Advancements in computer Science Engineering and Technology, International Journal on Scientific Research in Engineering(IJSRE), Volume: 1 Issue: 1, January 2017

[21] Ijteba Sultana, Dr. Mohd Abdul Bari ,Dr. Sanjay, "*Routing Quality of Service for Multipath Manets, International Journal of Intelligent Systems and Applications in Engineering"*, JISAE, ISSN:2147-6799, 2024, 12(5s), 08–16;

[22] Mr. Pathan Ahmed Khan, Dr. M.A Bari,: Impact Of Emergence With Robotics At Educational Institution And Emerging Challenges", International Journal of Multidisciplinary Engineering in Current Research(IJMEC), ISSN: 2456-4265, Volume 6, Issue 12, December 2021,Page 43-46

[23] Shahanawaj Ahamad, Mohammed Abdul Bari, Big Data Processing Model for Smart City Design: A Systematic Review ", VOL 2021: ISSUE 08 IS SN : 0011-9342 ;Design Engineering (Toronto) Elsevier SCI Oct : 021

[24] Syed Shehriyar Ali, Mohammed Sarfaraz Shaikh, Syed Safi Uddin, Dr. Mohammed Abdul Bari, "Saas Product Comparison and Reviews Using Nlp", Journal of Engineering Science (JES), ISSN NO:0377-9254, Vol 13, Issue 05, MAY/2022

[25] Mohammed Abdul Bari, Shahanawaj Ahamad, Mohammed Rahmat Ali," Smartphone Security and Protection Practices", International Journal of Engineering and Applied Computer Science (IJEACS) ; ISBN: 9798799755577 Volume: 03, Issue: 01, December 2021 (International Journal,U K) Pages 1-6

[26] .A.Bari& Shahanawaj Ahamad, "Managing Knowledge in Development of Agile Software", in International Journal of Advanced Computer Science & Applications (IJACSA), ISSN: 2156-5570, Vol: 2, No: 4, pp: 72-76, New York, U.S.A., April 2011

[27] Imreena Ali (Ph.D), Naila Fathima, Prof. P.V.Sudha ,"Deep Learning for Large-Scale Traffic-Sign Detection and Recognition", Journal of Chemical Health Risks, ISSN:2251-6727/ JCHR (2023) 13(3), 1238-1253

[28] Imreena, Mohammed Ahmed Hussain, Mohammed Waseem Akram" An Automatic Advisor for Refactoring Software Clones Based on Machine Learning", Mathematical Statistician and Engineering ApplicationsVol. 72 No. 1 (2023)

[29] Mrs Imreena Ali Rubeena,Qudsiya Fatima Fatimunisa "Pay as You Decrypt Using FEPOD Scheme and Blockchain", Mathematical Statistician and Engineering Applications: https://doi.org/10.17762/msea.v72i1.2369 Vol. 72 No. 1 (2023)

[30] Imreena Ali , Vishnuvardhan, B.Sudhakar," Proficient Caching Intended For Virtual Machines In Cloud Computing", International Journal Of Reviews On Recent Electronics And Computer Science , ISSN 2321-5461,IJRRECS/October 2013/Volume-1/Issue-6/1481-1486

[31] Heena Yasmin, A Systematic Approach for Authentic and Integrity of Dissemination Data in Networks by Using Secure DiDrip, INTERNATIONAL JOURNAL OF PROFESSIONAL ENGINEERING STUDIES, Volume VI /Issue 5 / SEP 2016

[32] Heena Yasmin, Cyber-Attack Detection in a Network, Mathematical Statistician and Engineering Applications, ISSN:2094-0343, Vol.72 No.1(2023)

[33] Heena Yasmin, Emerging Continuous Integration Continuous Delivery (CI/CD) For Small Teams, Mathematical Statistician and Engineering Applications, ISSN:2094-0343, Vol.72 No.1(2023)