A logo with a white background

Description automatically generated

# Department of Artificial Intelligence

## 22MAT121: **Discrete Mathematics**, Fall 2023

Project Report

# Handwriting Recognition for Authentication

**Team Members:**

Vishal Seshadri B: (CB.SC.U4AIE23260)

K S Venkatram (CB.SC.U4AIE23236)

Sanggit Saaran K C S (CB.SC.U4AIE23247)

Surya HA (CB.SC.U4AIE23267)

Harivaarthan T D (CB.SC.U4AIE23228)

**Supervised By:**

*Dr. Kritesh Gupta*

*Assistant Professor*

*Department of Artificial Intelligence*

*Amrita Vishwa Vidyapeetham*

Date of submission: 26.12.23

Signature of the Project Supervisor:

**Table of contents**

Chapters Page no.

1. Abstract…………………………………………………………………………….1
2. Introduction………………………………………………………………………...2
3. Literature review…………………………………………………………………...3
4. Future enhancements………………………………………………………………4
5. Conclusion…………………………………………………………………………5
6. References………………………………………………………………………….6

**ABSTRACT**

This research explores the integration of signature-based handwriting recognition as a novel approach for authentication purposes. Traditional methods of authentication often rely on passwords or biometric data, which may be susceptible to various security risks. This system provides a robust idea about how signatures are recognized using convolutional neural networks. The signatures are classified as forged and genuine and then trained accordingly. After recognition, we test our algorithm and find that we have an accuracy of 94.92%. we are using the SoftMax regression model for implementing the machine learning program.

**INTRODUCTION**

Handwriting recognition for authentication is a groundbreaking project that combines computer vision and machine learning. TensorFlow is used as a reliable framework to implement this project. Given the growing need for secure and efficient authentication methods, this project aims to develop a signature-based authentication system. Advanced techniques are employed to distinguish genuine signatures from forged ones. This will contribute to fraud detection and reduce the workload of manual verification. The project begins by creating a comprehensive dataset of diverse signatures that captures the inherent variability in individual writing styles. OpenCV is then used to process the signature images and extract relevant features, which form the basis of the machine learning analysis. Each signature is labelled as either genuine or forged, which provides a supervised learning framework for model training. TensorFlow's SoftMax regression model is used to train the system, allowing the model to learn the intricate patterns and dynamics within the signatures. The trained model's performance is then evaluated using distinct sets of signatures dedicated to testing, providing insights into its predictive accuracy. The project employs key evaluation metrics such as a confusion matrix and a Receiver Operating Characteristic (ROC) curve to assess the model's ability to classify genuine and forged signatures correctly. Although the authentication system offers a reliable means of distinguishing authentic signatures from forgeries, it is essential to acknowledge the inherent limitations of AI. The project emphasizes continual refinement and adaptation to evolving handwriting styles. The system authenticates signatures based on genuine samples while accounting for the possibility of occasional prediction errors. This project represents a significant stride towards enhancing security protocols through innovative applications of computer vision and machine learning. Automating the identification of fraudulent signatures, not only streamlines authentication processes but also contributes to the broader objective of minimizing manual intervention and safeguarding against potential security threats.

**4. EXPERIMENTS**

* 1. OpenCV:

The main idea of using OpenCV is to read the image and process the image in a

way that CNN will be able to utilize it.

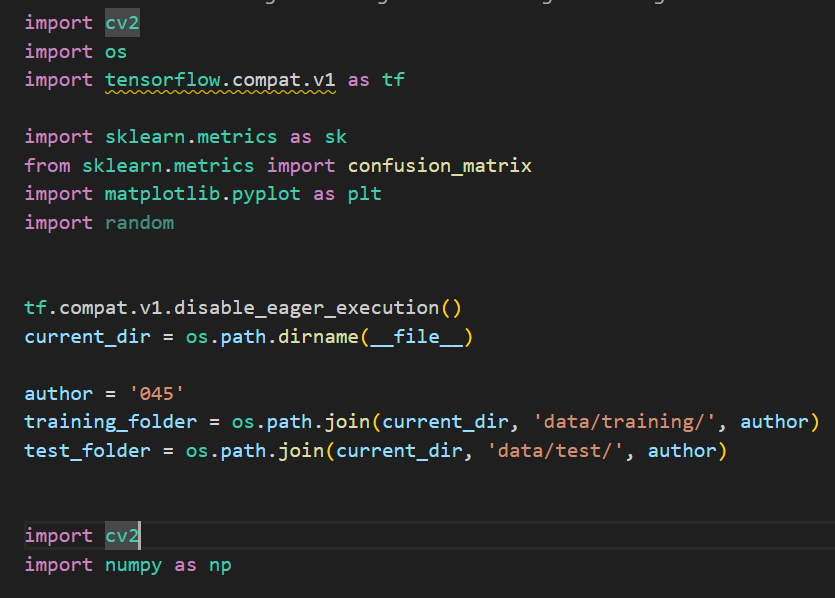
* 1. ****TensorFlow:  
     TensorFlow is an open-source machine learning framework developed by the Google Brain team. It is designed to facilitate the development and training of machine learning models, particularly neural networks. TensorFlow provides a comprehensive set of tools and libraries for building and deploying artificial intelligence applications.

Fig 4.1 code part 1

We are importing cv2 which is OpenCV and TensorFlow

Then we are importing os to get the current directory where the file with images has been stored

Sklearn. metrics is used for the confusion matrix and the roc curve which we are plotting with the help of matplotlib which is a built-in library in python.

Numpy is imported as the confusion matrix takes value from the NumPy library only.

* 1. Program Logic :

The below figure illustrates the steps involved in processing the images, preparing the images and converting them to pixel data and extracting the features to enable processing using tensors.

**Feature Extraction**

**Image Preparation**

**Resize**

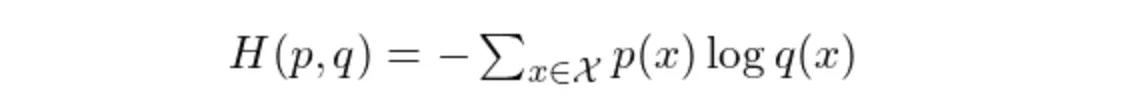
**Thresholding**

**Model**

**Parameters**

**Train**

**Cross Entropy**



**Optimize - GradDesc**

Fig 4.2 Processing Sequence

Image Preparation is done by using a function imageprep Since both gray and colour images can be used, the code checks to see if the image is a color image and then converts to gray, and using a denoising function clears any noise in the image to enable proper feature extraction.

Next we are using the threshold function Thresholding is a technique used to convert an input image into a binary image, where pixels are classified as either foreground or background based on a certain criterion. We are then inverting the threshold using binary invert

A black and white rectangular shapes

Description automatically generated

Fig 4.3 Threshold binary to binary invert

The parameters including the sum of columns and lines are extracted from the image binary and are appended to train.data for every image that is used in the training process. The below is the code segment that implements this image pre processing.

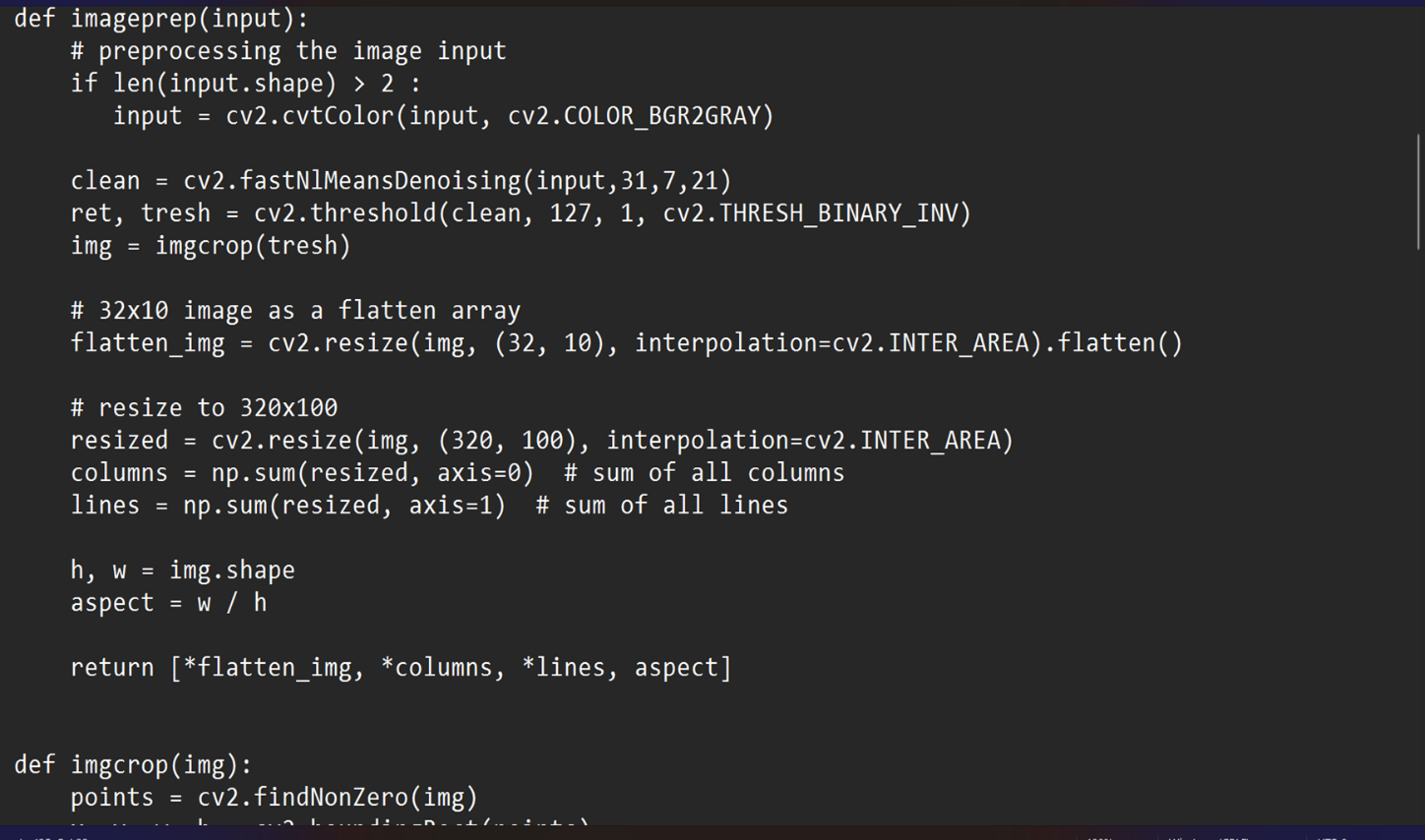


Fig 4.4 code part 2

The training data along with the label ( appropriate to the nature of the image – based on whether it is ‘geuine’ or ‘forged’ is stored . Once the image is processed and the data pertaining to that is extracted as binary and appended in the training tuple, the program continues to process the next image in the training set.

This is repeated for the test images as well.

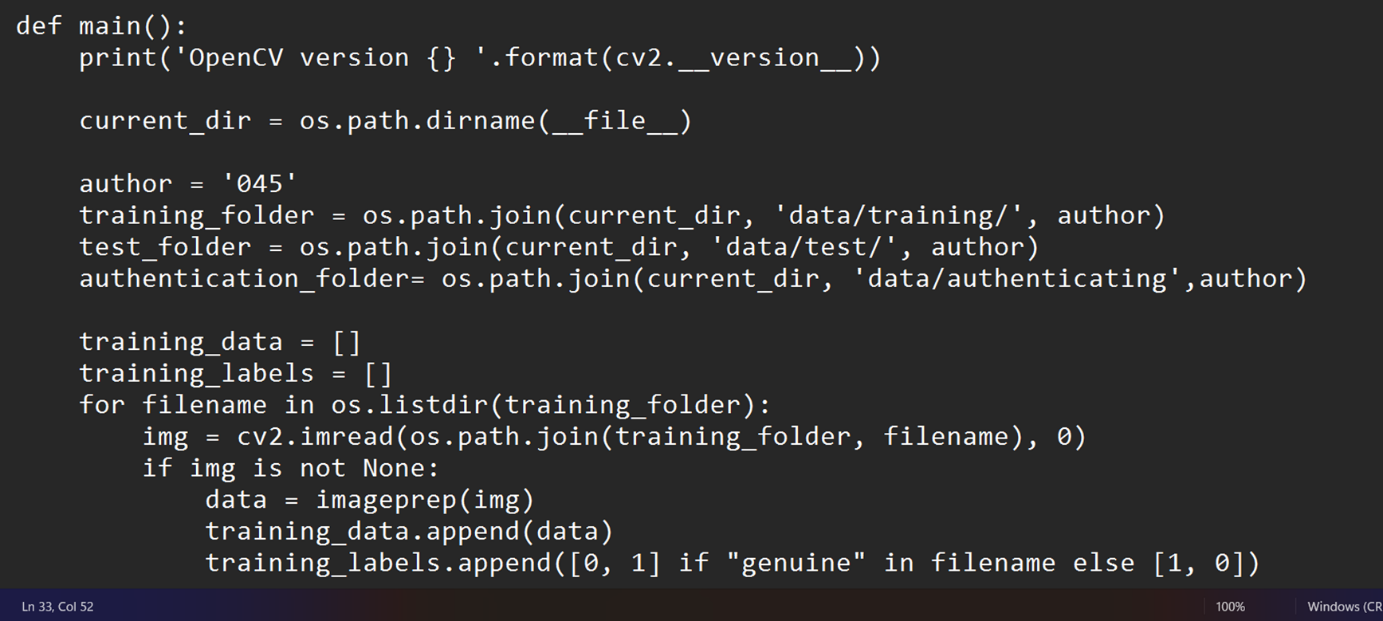


Fig 4.5 code part 3

Once the Test files are processed in this manner the training model is run using the runmodel function.

Here the model uses softmax regression to convert the binary data into probabilities. Softmax regression is widely used for image classification and hence the same has been used in our approach also.

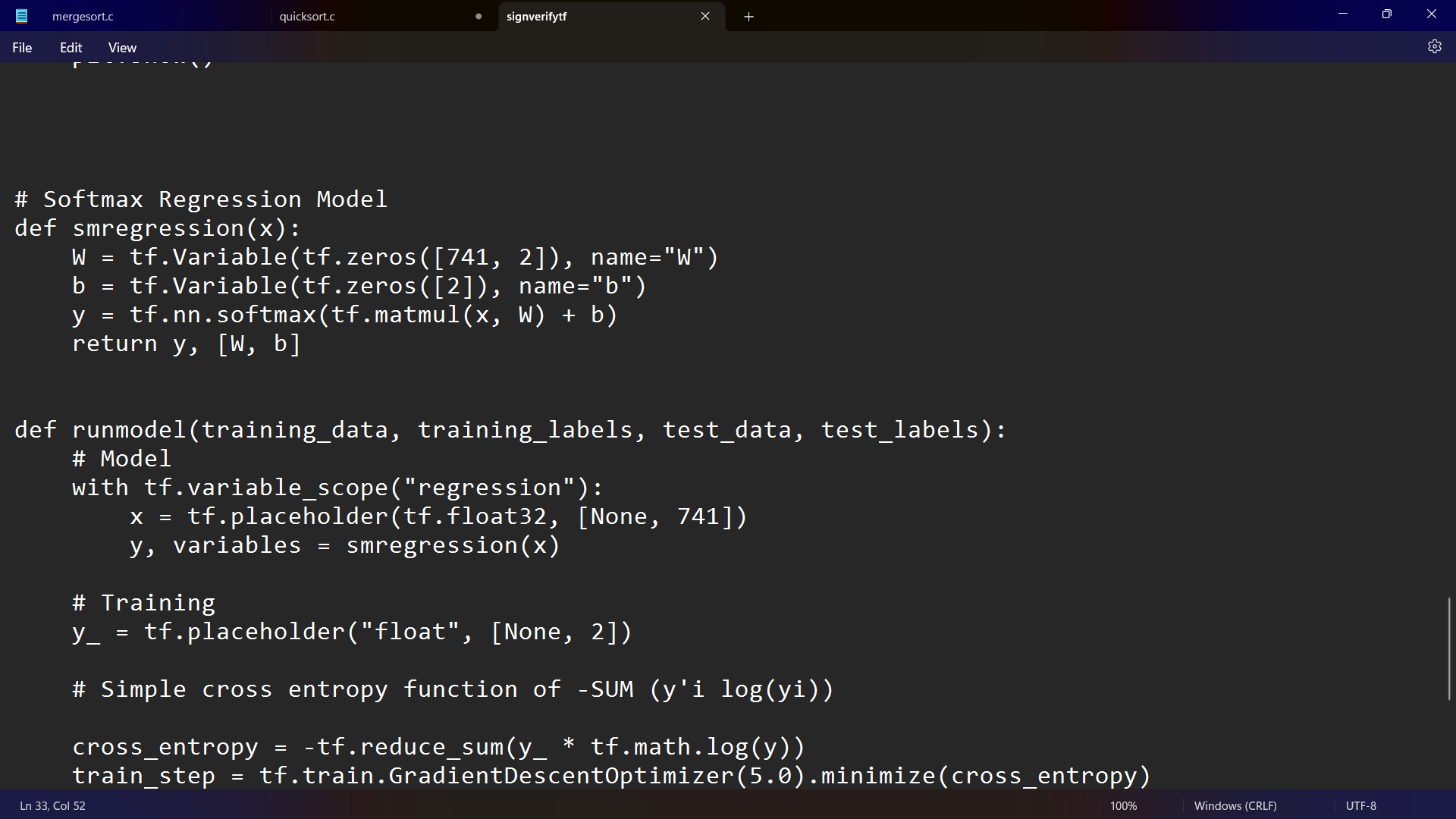
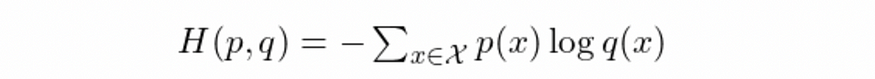


Fig 4.6 code part 4

To perform the training of the model we use a cross-entropy loss calculation function



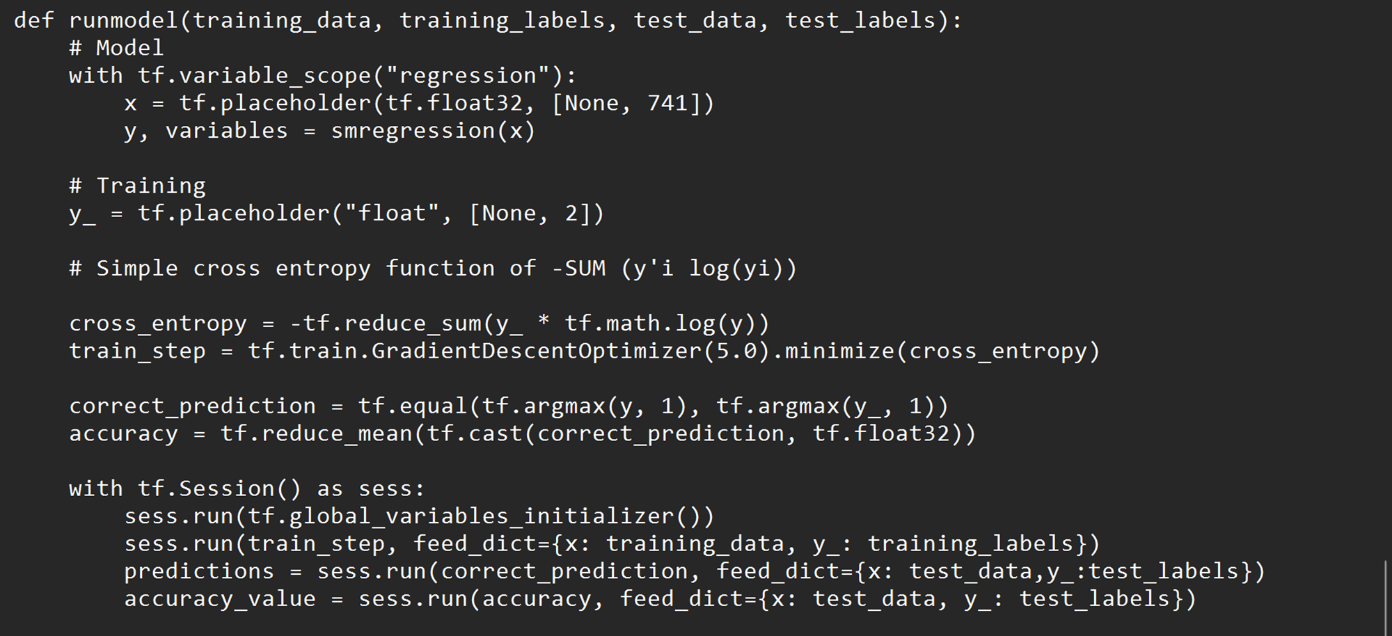
And the model is optimized using GradientDescentOptimizer function in tensorflow , while minimising the entropy cost.

Fig 4.7 code part 5

Now coming to the tf.session() where sess.run() function is used to run the entire training part of the TensorFlow. First all global variables are declared initially then the train\_step is given which contains the data that has been optimized using gradient descent.

The feed\_dict gives values for x and y where x takes the training data values and y\_ which has the placeholder takes the values of training labels.

Then the prediction is done using sess. run() where the correct prediction is used which contains the result whether the data is correctly predicted by our model or not.

Then finally we find the accuracy value of our model using the accuracy that we have already found out and using this we plot a confusion matrix and a roc curve

**LITERATURE REVIEW**

There has been a lot of research on handwriting recognition for authentication, for instance, Fang et al. [[1]](#one) proposed a handwriting detection tool based on incremental LSTM, wherein, they observed 99% accuracy in classifying the handwritten text.

In another study, Kurowski et al. [[2]](#two) utilized neural networks for classifying handwritten signatures. They reported that utilizing the triplet loss algorithm increased the predictive accuracy of the model.

One of the studies done by Ebrahimpour et al. [[3]](#three) used the pre-trained deep learning models for the detection of forged signatures and they compared both. The conclusion they have provided is that they have found that signature forgery detection has better accuracy. They have also used the Mobile Net model and found that it shows remarkable accuracy which went up to 98.44% and it seems it consumes less time than other models.

The next study done by Pansare A and Bhatia S et al. [[4]](#four) shows that they have used the idea of image processing, and neural networks for classifying signatures as forged and genuine. The conclusion they have made is about the accuracy which they have found to be 82.66%

The study done by Drott B. et al and Hassan-Reza T. et al. [[5]](#five) They have used the idea of max pooling and CNN to get the accuracy and find out the rate of false positive and true positive detection. They found that the false positive rate was about 0.6% and the true positive was 96.7%.

In the year 2020 Ahlawat S. et al, Choudhary A et al, Nayyar A et al, Singh S et al and Yoon, B et al. [[6]](#six) produced a project based on Handwritten digit using convolutional neural networks wherein they have presented a combination of learning parameters in designing a CNN for classifying the MNIST handwritten digits. They achieved a recognition accuracy of 99.87% for a MNIST dataset.

**CONCLUSION**

Handwriting authentication is a critical discipline, leveraging the precise individuality of anyone's writing style to verify their identification. With advancements in a generation, especially in machine-gaining knowledge of and pattern recognition, handwriting authentication strategies have substantially progressed in accuracy and reliability. These structures examine diverse capabilities consisting of stress, speed, and stroke patterns to create a comprehensive profile for authentication, making them an increasing number of strong. Moreover, handwriting authentication can be included in multifactor authentication systems, imparting an added layer of protection. By combining it with different biometric or knowledge-primarily based strategies, agencies can beautify standard security measures, reducing the chance of unauthorized right of entry. This makes handwriting authentication mainly precious in sectors consisting of finance, criminal documentation, and admission to control, wherein security is of extreme significance. Despite those improvements, demanding situations like variability in writing patterns and environmental factors persist. Continuous research and improvement are critical to address those challenges and in addition, improve the reliability and robustness of handwriting authentication systems. As the era evolves and methodologies refine, destiny holds promise for even more sophisticated and stable handwriting authentication systems, contributing to typical improvements in the subject of biometrics and safety. Handwriting authentication, harnessing the individualistic nature of anyone's writing, has emerged as an increasing number of dependable and accurate because of technological advancements. Its integration into multifactor authentication systems enhances safety features, particularly in sectors like finance and admission to manipulation. However, demanding situations inclusive of variability in writing styles persist, necessitating ongoing studies and improvement to enhance the reliability and robustness of handwriting authentication systems, paving the manner for even greater sophisticated and stable methodologies in the future. This project could be improved in several ways. Firstly, a graphical user interface could be provided to display which signatures are being authenticated and which ones are not. Secondly, handwriting recognition could be used instead of signatures as they can be misread, and two people may have identical signatures but distinct handwriting. Advancements in deep learning algorithms and artificial intelligence can significantly improve the accuracy and complexity of handwriting analysis. Neural networks are particularly useful for capturing subtle nuances in writing styles, making authentication systems more reliable.

**REFERENCES**

[1] Fang, L., Zhu, H., Lv, B., Liu, Z., Meng, W., Yu, Y., Ji, S. and Cao, Z., 2020. HandiText: Handwriting recognition based on dynamic characteristics with incremental LSTM. *ACM Transactions on Data Science*, *1*(4), pp.1-18.

[2] Kurowski, M., Sroczyński, A., Bogdanis, G. and Czyżewski, A., 2021. An automated method for biometric handwritten signature authentication employing neural networks. Electronics, 10(4), p.456.

[3] Ebrahimpour, N., HANDWRITTEN SIGNATURES FORGERY DETECTION USING PRE-TRAINED DEEP LEARNING METHODS.

[4] Pansare, A. and Bhatia, S., 2012. Handwritten signature verification using neural network. International Journal of Applied Information Systems, 1(2), pp.44-49.

[5] Drott, B. and Hassan-Reza, T., 2015. On-line handwritten signature verification using machine learning techniques with a deep learning approach. Master's Theses in Mathematical Sciences.

[6] Ahlawat, S., Choudhary, A., Nayyar, A., Singh, S. and Yoon, B., 2020. Improved handwritten digit recognition using convolutional neural networks (CNN). Sensors, 20(12), p.3344.