A DEEP LEARNING- BASED APPROACH TO VERIFY PUNJABI SIGNATURE

**DISSERTATION**

MASTERS OF TECHNOLOGY

IN

COMPUTER SCIENCE & TECHNOLOGY

(COMPUTER SCIENCE AND TECHNOLOGY)

**Supervised By:**

Er. Surinder Singh Khurana

Assistant Professor

Dept. Comp. Sci. & Tech

**Submitted By:**

Nancy Mittal

M.Tech in Computer science and tech.

Reg. No. 20mtcsnt14

Dept. Computer Science & Technology

**Department for Computer Science & Technology**

**School of Engineering & Technology**

**Central University of Punjab**

**Bathinda-151401**

**JUNE 2022**

# **DECLARATION**

I declare that the dissertation entitled “**A DEEP LEARNING- BASED APPROACH TO VERIFY PUNJABI SIGNATURE**” has been solely prepared by me under the guidance of Er. Surinder Singh Khurana, Assistant Professor, Centre for Computer Science and Technology, School of Engineering and Technology, Central University of Punjab, Bathinda. No part of this dissertation has been formed as the basis for the award of any degree or fellowship previously.

Nancy Mittal

Registration No. 20mtcsnt14

Centre for Computer Science and Technology

School of Engineering and Technology

Central University of Punjab,

Bathinda-151001

Date:

# **CERTIFICATE**

I certify that Ms. Nancy Mittal has prepared his dissertation entitled “**A DEEP LEARNING- BASED APPROACH TO VERIFY PUNJABI SIGNATURE**”, for the award of Master of Technology in Computer Science and Technology from Central University of Punjab, under my guidance. She has carried out his research work at the Centre for Computer Science and Technology, School of Engineering and Technology, Central University of Punjab, Bathinda.

Er. Surinder Singh Khurana

Assistant Professor

Centre for Computer Science and Technology,

School of Engineering and Technology,

Central University of Punjab,

Bathinda-151001

Date:

# **ABSTRACT**

**“A DEEP LEARNING- BASED APPROACH TO VERIFY PUNJABI SIGNATURE”**

|  |  |
| --- | --- |
| Name of the Student: | Nancy Mittal |
| Registration Number: | 20mtcsnt14 |
| Degree for which Submitted: | M.Tech in Computer Science and Technology |
| Name of the Supervised: | Er. Surinder Singh Khurana |
| Name of Department: | Computer Science & Technology |
| Name of School: | School of Engineering & Technology |
| Keywords: | CNN, DenseNet201, Resnet50, offline  Signature verification. |

Offline Punjabi Signature verification framework faces many challenges because of the complexities in the writing pattern in the language. Limited research was done due to Punjabi signature dataset is not available. So we have collected the dataset from the 572 authors. There is one document sheet for the each author in which there are 15 forged and 5 genuine signatures. So there are total 11440 samples of the signatures in which there are 2560 genuine signatures and 8580 forged signatures. For the verification of the signatures we have implemented 3 models named as CNN, ResNet50 and DenseNet201 in which convolutional layers are different in number. The More accurate results obtained by the model ResNet50 that predicts the verification with 88.04% accuracy and the DenseNet201 and CNN with 53.40% and 67.31% respectively.

|  |  |
| --- | --- |
| **Nancy Mittal** | **Er. Surinder Singh Khurana** |

# **ACKNOWLEDGMENT**

I would like to express my sincere gratitude to my research supervisor, Er. Surinder Singh Khurana, Assistant Professor, Department of Computer Science and Technology, for providing his valuable guidance and suggestions. I would specially thank him for constantly motivating me throughout the course of the Dissertation.

I would like to thank Dr. Amandeep Kaur, Associate Dean, School of Engineering and Technology, for providing us an opportunity to work in the domain of image preprocessing through this Dissertation.

I would also like to thank my parents and all my friends who were a constant source of strength for me.

Nancy Mittal

**TABLE OF CONTENT**

[*DECLARATION ii*](#_Toc106187999)

[*CERTIFICATE iii*](#_Toc106188000)

[*ABSTRACT iv*](#_Toc106188001)

[*ACKNOWLEDGMENT v*](#_Toc106188002)

[*LIST OF FIGURES viii*](#_Toc106188003)

[*LIST OF TABLES ix*](#_Toc106188004)

[CHAPTER 1 INTRODUCTION 10](#_Toc106188005)

[1.1 Punjabi languages 12](#_Toc106188006)

[1.2 Features of Punjabi Language (Gurmukhi) 12](#_Toc106188007)

[1.3 Machine Learning Classifiers 14](#_Toc106188008)

[1.3.1 Deep Learning 14](#_Toc106188009)

[1.3.2 Convolutional Neural Networks 14](#_Toc106188010)

[1.5 Objectives 20](#_Toc106188011)

[CHAPTER 2 LITERATURE REVIEW 21](#_Toc106188012)

[2.1 Literature Review 21](#_Toc106188013)

[2.2 Summary of Literature Review 27](#_Toc106188014)

[2.3 Knowledge gap 33](#_Toc106188015)

[CHAPTER 3 METHODOLOGY 34](#_Toc106188016)

[3.1 Dataset Collection 35](#_Toc106188017)

[3.1.1 Digitization 36](#_Toc106188018)

[3.1.2 Box Detection and Cropping 36](#_Toc106188019)

[3.2 Preprocessing 37](#_Toc106188020)

[3.2.1 Training Data and Testing Data 40](#_Toc106188021)

[3.2.1.1 CNN 40](#_Toc106188022)

[3.2.2 CNN Parameters 42](#_Toc106188023)

[3.4 Performance Evaluation 43](#_Toc106188024)

[CHAPTER 4 RESULTS AND DISCUSSION 44](#_Toc106188025)

[4.1 Results obtained by ResNet50 44](#_Toc106188026)

[4.2 Results Obtained by the DenseNet201 46](#_Toc106188027)

[4.3 Results Obtained by the CNN 47](#_Toc106188028)

[4.4 Comparison with existing systems 48](#_Toc106188029)

[CHAPTER 5 CONCLUSION AND FUTURE SCOPE 49](#_Toc106188030)

[5.1 Conclusion 49](#_Toc106188031)

[5.2 Future work 49](#_Toc106188032)

[REFERENCES 50](#_Toc106188033)

# **LIST OF FIGURES**

[Figure 1 Punjabi language numerals from 0 to 9 13](#_Toc105745794)

[Figure 2 Punjabi language vowels and consonants 13](#_Toc105745795)

[Figure 3 Convolution operation with a kernel size 3x3. 15](#_Toc105745796)

[Figure 4 ReLU activation function 16](#_Toc105745797)

[Figure 5 average pooling 18](#_Toc105745798)

[Figure 6 Max Pooling 18](#_Toc105745799)

[Figure 7 CNN’s fully connected layer 19](#_Toc105745800)

[Figure 8 overall methodology 34](#_Toc105745801)

[Figure 9 document design 35](#_Toc105745802)

[Figure 10 Handwritten document into electronic format 36](#_Toc105745803)

[Figure 11 Individual Signature 37](#_Toc105745804)

[Figure 12 Res Net architecture 41](#_Toc105745805)

[Figure 13 ResNet50 Model 41](#_Toc105745806)

[Figure 14 DenseNet201 Model 42](#_Toc105745807)

[Figure 15 Results obtained by Resnet50 44](#_Toc105745808)

[Figure 16 Accuracy Graph 45](#_Toc105745809)

[Figure 17 Loss in Model 45](#_Toc105745810)

[Figure 18 Confusion Matrix of ResNet50 45](#_Toc105745811)

[Figure 19 Results obtained by the DenseNet201 46](#_Toc105745812)

[Figure 20 Confusion Matrix of DenseNet201 46](#_Toc105745813)

[Figure 21 ROC Curve by the CNN 47](#_Toc105745814)

[Figure 22 Difference Scores In the images 47](#_Toc105745815)

# **LIST OF TABLES**

[Table 1 Summary of Results of offline signature verifications 27](#_Toc105748120)

[Table 2 Description of the datasets 37](#_Toc105748121)

[Table 3 Pre-Processing Steps 38](#_Toc105748122)

[Table 4 Confusion Matrix 43](#_Toc105748123)

[Table 5 Losses and accuracies of different models 44](#_Toc105748124)

[Table 6 Precision, recall and F1 score of ResNet50 45](#_Toc105748125)

[Table 7 Precision, recall and F1 score of DenseNet201 46](#_Toc105748126)

[Table 8 Precision, recall and F1 score of CNN 47](#_Toc105748127)

[Table 9 Comparisons with the existing techniques 48](#_Toc105748128)

# **CHAPTER 1 INTRODUCTION**

With the increase in digitization in every field, finances and business, signatures have become the most critical part. Hence we need different approaches for authentication and verification. The duplicate signature may be signed by the fraudster who knows about the target user's signature for breaking the system in the real world. In the end case, the fraudster does not know the target's name and signature, and in the intermediate points, the fraudster may know the title but not the sign's shape. In real world, Saikia & Chandra Sharma (2012) defines the forgery into three types: random forgery, simple forgery, and skilled forgery

*Random Forgery*: - In the random forgery, the forger doesn't have any data about the user's name and instead uses his signature, and the shape is entirely different from the user's signature.

*Simple Forgery*: - In simple forgery, only the user's name was known but not the shape, and the signature shape may be similar to the user's signature.

*Skilled Forgery*: - In the skilled forgery, both the user's name and signature with some rehearsing to perform a good faking for the user's signature.

In Pawar, (2015) research, they proposed that there are two types of signature identification and verification systems: offline and online systems.

*Offline systems*: - In the offline systems, the signature is scanned from any document such as bank checks, UIDAI offline e-KYC.

*Online Systems*:- In the online system the devices are required like tablets and smartphones to capture various other information such as time, pressure, pen up and pen down; for example, in Flip kart delivery services, we have to do signature on the tablets for the acknowledgment.

In the signature identification system, user's signature is given to the system, then the system equates the signature with all other signatures enrolled in the dataset and analyses the similarity results. There are two approaches for signature verification: writer dependent and writer independent. Saleem & Kővári, 2019; et al., (2020) compare the techniques and description is given below.

*Writer Independent*: - In the writer independent, one model is trained for all users and manages the matching of the query to reference signatures in space. Most researchers prefer this approach. The feature learning from the signature images in the writer independent format is known as SigNet. After the processing is being done then the feature extraction is done to extract features, which further uses the writer dependent approach for each writer using various machine language classifiers known as Support vector machines. There is significant approach for the improvement of the performance.

*Writer dependent*: - In the writer dependent, a specific model is trained for each user and is responsible for authenticating their signatures, so the system needs to retrain when adding a new signer.

The system's performance was checked by the False Rejection Rate of genuine signatures and the False Acceptance Rate of the forgery signature. There is an equal error rate, the error rate where both the FAR and FRR are similar, or the Distinguishing Error Rate. Described EER can demonstrate the DER.

We are using a Convolutional neural network (CNN) based validation system in this research work. Some related research is Object recognition, computer vision, and speech recognition with the help of some deep learning methods to get better datasets, data enhancement, additional feature extraction methods, and data augmentation methods. We tackle the CNN-based system ResNet50 to verify the offline writer's independent signature. We are compiling our model using Adam optimizer and binary cross entropy loss function. As we research the Punjabi signatures. Research aims to provide the data so that more research should be done on the Punjabi dataset using various models. We have done preprocessing of the datasets, which extracted different features. We will generate batches to create pairs of authentic genuine and genuine-forged images. For every person, we have 20 signatures; 5 genuine and 15 forged signatures. To know more about the characterization we will study the Punjabi language in the next section.

## **Punjabi languages**

Punjabi is an Indo-Aryan language spoken by the Punjabi people in Pakistan and India. Punjabi evolved from Prakrit and afterwards Apabhramsa languages. Punjabi is one of India's 23 official languages. According to the 2011 Indian census, 31.14 million people identified Punjabi as their first language, ranking it the world's ninth most extensively 18 spoken language and India's eleventh. Punjabi is the third most widely spoken language in England and the fifth most widely spoken language in Canada. It is also spoken as a 19 minority language in a number of other countries, including the United States, Australia, the United Kingdom, and Canada, where large percentages of Punjabis have immigrated. Gurmukhi, a Brahmic script, is used by Indian Sikhs. The two scripts in Punjabi are known as Gurumukhi and Shahmukhi in India and Pakistan, respectively. Punjabi is written in "standard orientalist" transcription.

Subscript dots imply retroflex consonants, macrons suggest etymologically contrastively long vowels, & denotes aspirated stops. Low and high tones are indicated by tildes, whereas nasalized vowels are indicated by grave and acute accents. Several scripts are used to write Punjabi, depending on area, dialect, and religion of both the speaker Shahmukhi (from the mouth of the Kings) is a modified variant of the Persian-Nasta'liq (Arabic) script used in Pakistan's Punjab province. Sikhs and others in the Indian state of Punjab use the Gurmukhi script (from the mouths of the Gurus). Devanagari is a script used among Hindus and residents of surrounding Indian states such as Haryana and Himachal Pradesh to write 12 Punjabi. However, Gurmukhi and Shahmukhi scripts are the most commonly used for writing Punjabi and are considered the official scripts of the language.

## **1.2 Features of Punjabi Language (Gurmukhi)**

There is no abstraction of upper and lowercase letters.

Unlike Greek and Roman alphabets, the Gurmukhi script is arranged logically, featuring vowels occurring first, followed by consonants, and finally semi-vowels.

All consonants have a built-in vowel in this syllabic script. Diacritics are used to enhance the built-in vowel and can exist above, below, before, or after the consonant.

Vowels are written as individual letters when the Gurmukhi letters come in the beginning of a syllable. Special conjunct symbols combine the monitor its performance of each letter when determining consonants that occur together. Figure 1 depicts the numerals, whereas Figure 2 depicts the vowels and consonants.

**੦ = 0, ੧ = 1, ੨ = 2, ੩ = 3, ੪ = 4, ੫ = 5, ੬ = 6, ੭ = 7, ੮ = 8, ੯ = 9**

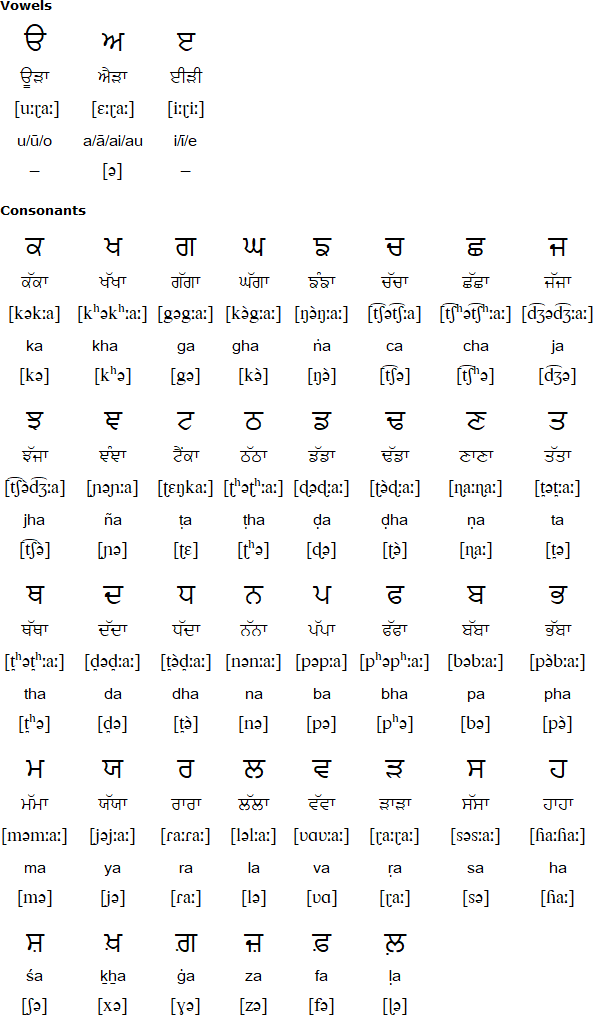
Figure 1 Punjabi language numerals from 0 to 9

Figure 2 Punjabi language vowels and consonants

## **1.3 Machine Learning Classifiers**

The classification is a work of assigning the labels to the datasets in the problem. We can use many algorithms in machine learning for other categories such as Support Vector Machine, Naïve-Bayes, k-nearest neighbors, neural networks, deep learning, and Convolutional Neural Networks (CNN). In research work we are using the Convolutional Neural Networks (CNN) which is the part of deep learning. In the next section, we are briefly discussing deep learning and CNN.

### **1.3.1 Deep Learning**

According to the research done by Alzubaidi et al., (2021) the deep learning is the subset of machine learning. The main base is inspired by the human-brain. It uses various algorithms such as artificial neural networks. The various classification steps involved are pre-processing, feature extraction, wise feature selection and learning and classification. Incorrect discrimination between different classes is known as biased feature selection. We are using deep learning approach because it performs in all application domains. In the automated manner the optimized features are learned. There are various types of neural networks: RNNs, CNNs. As CNN is discussed in the next part.

### **1.3.2 Convolutional Neural Networks**

The Patil & Rane, (2021) studied that the convolutional neural networks are the subset of deep learning techniques prevalent in computer vision tasks. CNN is designed instinctively and adaptively to learn spatial hierarchies of features, from low to high level patterns. CNN is a mathematical design of three building blocks: convolution, pooling, and fully connected layers. The first two layers perform feature extraction, and the third extracts features into a final output such as classification and composes a mathematical operation: stack. In the images, pixel values are stored in a 2D grid in the form of an array of numbers and kernels are a small grid of parameters, to each image position an optimized feature extractor is applied. One layer sustains its output into the next layer. Training is the optimizing parameters, which minimize the difference between outputs and ground truth labels through an optimization algorithm called back-propagation and gradient descent. The details of the CNN is researched by the Mushtaq et al., (2021)

* **Convolution Layer**

A CNN categorizes an image in terms of spatial properties like edges, strokes, contours, textures, gradients, and orientation, representing it in a different layer. It combines linear (convolution) and nonlinear (activation function) operations to perform feature extraction. In a convolution operation, a small array of numbers is applied to an input array of numbers by performing the summation of the element-wise product of each kernel element and input tensor to obtain the output value in the position of output tensor called as a feature map as shown in (figure 3)

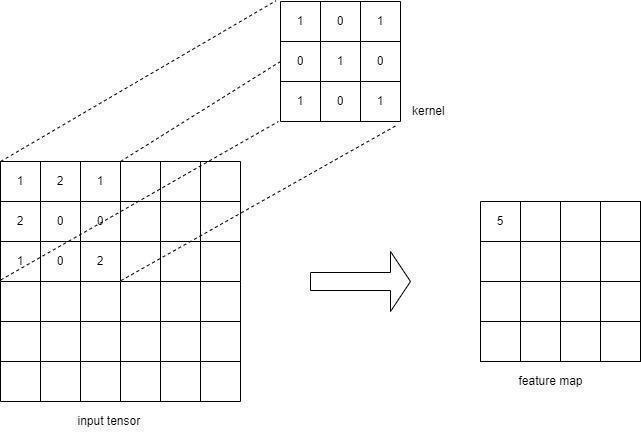


Figure 3 Convolution operation with a kernel size 3x3.

Various characteristics of the input tensor are represented by different kernel for generating an arbitrary number of feature maps. Thus, different kernels can be treated as other feature extractors of layers. Basically there are two main hyper-parameters: size and number of 48 kernels in which the size of kernel is a tuple of 3x3 but sometimes 5x5 or 7x7. At the same time, the number of kernels is inconsistent and determines the depth of the output feature maps.

Two other convolutional operation parameters: padding and strides: Padding can be 'same' on 'valid.' Setting padding to 'valid' means that the spatial dimensions of the output feature map are permitted to reduced and compared to the input tensor via the natural application of the convolutional because it does not allow the center of each kernel to overhang the outermost element of an input tensor. A valid successive after the convolutional operation feature map would get smaller. ‘Same’ padding is used when we continue spatial dimensions of the output feature map with the input tensor. To keep them in same plane dimension, zeros are added in matrix form to the input tensor so as to fit the center of kernel on the outermost element during the convolutional operation. Strides are the distance between the two successive kernel positions.

In other words, strides denote the number of pixels, the kernel window moves after each functional operation. The characteristic choice of a stride is one. The CNN model processes the kernel automatically for a given training dataset with the hyperparameters such as the number of kernels, size of kernels, padding, and stride.

* **Non-Linear activation function: ReLU**

Once the convolved outputs are obtained, they are passed through a non-linear activation function for non-linear transformation. Different non-linear activation functions include sigmoid, tanh, and rectified linear unit (ReLU). ReLU is less computationally expensive than tanh and sigmoid because it involves similar mathematical operations, and only a few neurons are activated at a time, making the networks efficient and easy for computation. In simple words, ReLU learn much faster than tanh and sigmoid function and simply compute the function: f(x) = max (0, x). (See Figure 4)

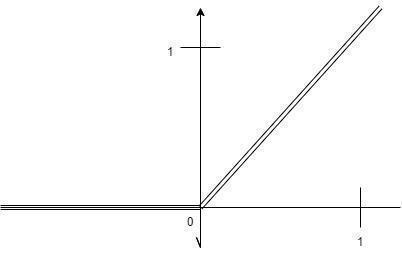
**

Figure 4 ReLU activation function

A neural network generally updates the weights and biases of the neurons based on output error. This process is called back-propagation. The back-propagation is the only possible activation function from the gradient supplied and the error to update the weights and biases.

* **Pooling layer**

A pooling layer is a basic downsampling procedure that reduces the size of each temporal dimension of output feature maps, lowering the number of subsequent learnable parameters and introducing translation invariance to tiny shifts and distortions. Pool size, stride, and padding are the hyper-parameters in pooling processes, much like they are in convolution. However, neither of the pooling layers have such learnable parameters.

* **Max pooling and Average pooling**

Each pooling Operation in max pooling collects patches from the input feature maps and then determines the current patch's maximum value. Average pooling, implies that the average of all the parts in the most recent patch, as demonstrated in (figure 5) and (figure 6). The in-plane dimensions of the feature map were downsampled by a factor of 2 using a max-pooling or average pooling method and the depth dimension of the feature maps stays intact.

* **Global Max pooling and Global average pooling**

By picking the maximum value or taking the average of all the components in each feature map, the global max pooling or global average pooling procedure downsamples the in-plane dimensions of a feature map into a 1x1 array, but the depth of the feature map is preserved. These pooling processes decrease the amount of learnable parameters, allow the CNN to take variable-size inputs, and are only used once before the fully linked layers.

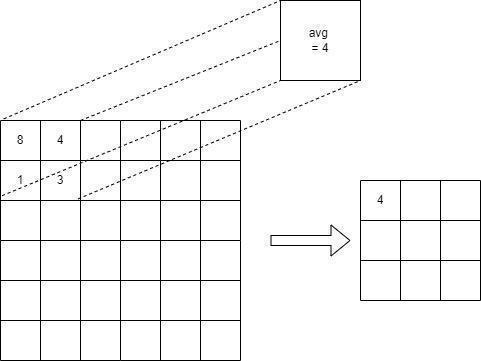


Figure 5 average pooling

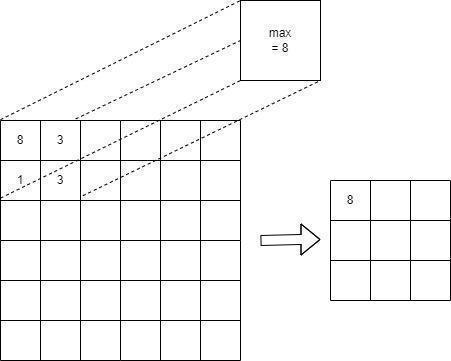


Figure 6 Max Pooling

* **Fully connected layer**

Fully connected layers use a flattened input, in which a portion of the fully connected layer maps each input to the network's final output as the projected probabilities of each class for final classification, as illustrated in (figure 7). In other words, the last convolutional or pooling layer's output feature maps are flattened and linked to one or more fully connected layers, also known as dense layers, in which each input is coupled to each output by a learnable weight. The technique of flattening involves converting multidimensional feature maps into a one-dimensional array of numbers. The number of output nodes in the final dense layer is usually equivalent to the number of classes. Each completely linked step is represented by a nonlinear function.

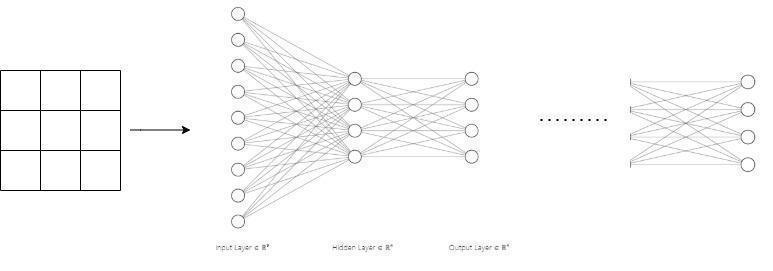


Figure 7 CNN’s fully connected layer

* **Last layer activation function**

The softmax function is an extended logistic function that accepts a set of scores as input and returns a set of probabilities as output. In other respects, a softmax function is a multiclass classification activation function that normalizes the output of the final dense layer to target class probabilities, where each value lies in the range 0 to 1, and all values aggregate to 1.

**1.4 Problem Statement**

The literature review has a guide to identifying several problems processed and verified by the various authors, and multiple techniques are applied with different accuracy. The datasets are selected for processing the GPDS series and CEDAR. But limited research is done in Punjabi (Gurmukhi) script. The first issue is due to the lack of writing patterns. The second issue is the shape of the words because their position where they have been written is also an issue that implies difference. Another reason and the problem are that the dataset is not publicly available, which stops the research (Although various studies have been done in English, Devanagari, Persian, and many more but in Punjabi). Most of the work is done in HMM, SVM, Euclidean distance, and KNN, and less in neural networks. The various techniques that work towards feature extraction, segmentation, classification, user-based score normalization, and recognition patterns provide better performance. Thus, the field is new experimentation towards Punjabi and Punjabi script, which provide better accuracy results and generalized image recognition.

## **1.5 Objectives**

1. To collect the handwritten signature dataset from different authors.
2. To extract the features from the dataset.
3. To evaluate the performance with deep learning to recognize signatures.

# **CHAPTER 2 LITERATURE REVIEW**

## 

## **2.1 Literature Review**

Offline signature verification is a well-researched topic; some different approaches and techniques have been studied. All the prevalent papers related to the offline signature verification using different datasets are included in this section. It can help the researchers in the Punjabi contour-based system field to study the classifiers and accuracy of the techniques. Here we review some of the research on offline signature verification.

**2.1.1. Locating the region of interest:** - To verify any signature firstly we extract the signature from the document. The original records are present where the signatures are available in real life. The region is extracted then verification is done. The relation between the handwriting and signature is analyzed by Ru et al. The fractal dimension measures the irregularity or fragmentation of a set. This helps in extracting the different properties related to writing and signature. The various properties are cursive handwriting, legible writing, and separate writing. This represents the difference between when the person writes and when they sign. The resources for gaining the information are different because both use independent identifiers. Ru et al’s research, signature region extraction work is done and it contains 350 documents signed by 70 different persons who have Persian or Arabic cursive signatures. The matter is present in various Arabic, Persian, and English alphanumeric with different fonts and sizes, a company logo, some horizontal and vertical lines, and cursive signatures. The signature is found with 98.86% and extracted with an accuracy of 97.71%.

A localization in scanned documents was proposed by celoğlu & Oğul (2014). The framework distinguishes the classification of segmented image regions using a two-phase connected labeling approach over an SVM classifier. A banking system, a framework, is used because of the good achievable accuracy to be used in the application.

**2.1.2. Determining the signature type:** The properties are determined by the end, beginning, change in shape, or bounding box. They are distinguished based on the complexity of the signature, which mainly depends upon the trajectory length and overlap. Alonso-Fernandez et al., (2007), proposed that the users are classified according to the types of signatures as a simple flourish, complex flourish, simple flourish with name, and complex flourish with name. The distribution of users in the MCYT-corpus is found as C1 (6.67%), C2 (17.33%), C3 (46.67%), and C4 (29.33%). EERs are arranged in ascending order as C4, C2, C3, and C1. The due result is drawings that make the signature harder, and adding more information is more challenging.

Pal et al., (2013), researched the multi-script signature of Bengali (Bangla), Hindi (Devanagari), and English for the identification process. The classifier for signature identification is SVMs. A database of 2100 Bangla signatures, 2100 Hindi signatures, and 2100 English signatures obtained an accuracy of 92.14% by the gradient feature using 4200 samples for training and 2100 samples for testing. This approach is used in real-life scenarios for signature language identification.

**2.1.3. Robustness to variations**: Nguyen et al., (2009), studied that a genuine signature may vary in illumination, rotation, translation, scaling, pen thickness, and noise. The alignment of two signatures may be altered and the basic global features extracted from the full signature. The variations are due to the rotation, translation, and scale variations.

The local derivative pattern gives the result with the accuracy of 15.35% using the GPDS-300 database as a superset of GPDS- 160. With the addition of the maximum noise level, the EER was 16.43%.

**2.1.4. Features**: There are various features in offline signature verification. In the instance method, local shape descriptors are used by Sabourin, (1997). This represents the conics of the handwritten signature, which simplifies the signature by Bortolozzi et al., (1997). The simplification doesn't enhance the verification success, but it is used for the verification in random forgery.

*Shape matrices* are also proposed by Sabourin, (1997) in which the evaluation is done in various steps. First, evaluate the object's centroid and then the pattern's primary orientation in the 2D space. In offline signature, the baseline is the natural choice for choosing the patterns and operations accomplished by the statistical moments. Accordingly, the invariance in translation and orientation is obtained. The next step is to locate the circumscribing circle of the pattern in which the binary shape matrices are accepted as a straightforward way of measuring the similarity between the matrices. The DER is 0.84% on a private database for testing where random forgery is done.

*Radon transform* is proposed by Coetzer et al., (2004), which extracts the features for feeding the HMM. The HMMs train feature extraction from local regions (local features) and in whole (global features). Each signature is zoned into some coinciding circular retinas where features are extracted from the radon transform. In the global retina, the whole retina is studied.

The fuzzy model was proposed by Hanmandlu et al., (2005) in which they studied the *distance distribution and angle distribution* of the image partition. The different samples have different variations in the same features. The fuzzification is done by the exponential membership function in the T.S. model as the structural parameters. The structural parameters talk about the variations due to handwriting styles and reflection of moods. The optimization of the model yields the solution for the parameters. The two T.S. models have a rule for each input feature in the first step and a single rule for all the input features. The T.S. a model with many regulations performs better accuracy than the T.S. model with single rules.

Igarza et al., (2005) presented that the Left-to-Right HMMs (LR-HMM) were utilized in extended models in static or offline signature processing using image connectivity analysis. The chain encoding of perimeter points for each bob obtained by research is an ordered set of points in the space, clockwise around the perimeter. Two models were created; in the first one, the blobs were ordered consequently by perimeter length, and in the second, the natural reading order was there, i.e., from top to bottom and left to right. So both the models were trained using the (x, y) coordinates and local geometrical features such as polar coordinates as center of ink, local radii, segment lengths, and local tangent angle. The EER was 27.58% in the MCYT baseline corpus.

*Contour features* were studied by Gilperez et al., (2008). Contour-Direction probability distribution function (PDF) represents histogram of angles, Contour-hinge PDF, Direction Co-Occurrence PDFs, Run-Length PDFs. For comparing the PDFs of a query and a reference, χ 2 metrics are used. The individual features, i.e., the mean value of the Hamming distances, are used as similarity metrics. The best-performing PDFs were Contour-Hinge PDF, with EER 10.18% and five genuine signatures, utilizing the MCYT corpus.

Larkins & Mayo (2008), studied that the gradient direction and *equi-mass spatial pyramids* features are extracted. Adaptive feature thresholding is the person-dependent offline signature verification method. In this, the simple image features are converted to binary feature vectors. Then the similarity was easily computed. The work was performed on the dataset GPDS-39 with the DER of 14.01% with the 12 references.

*Local Interest Points*, the local scale maxima, are the scale-space representation. The characterization of local neighborhoods of descriptors is calculated using the scale invariant feature transform (SIFT). The similarity between reference and query signatures descriptors is generalized using the baseline methodology. The Bayes classifier obtains the final result. The dataset used is the GPDS-160 signature dataset and DER is 15.3% by Ruiz-Del-Solar et al., (2008).

*High-Pressure Points*: - The information about the pressure distribution from static images of handwritten signatures was analyzed for verification. From grayscale images, we obtain histograms and calculate pseudo-spectral coefficients. The unique minimum phase is estimated, and feature vectors are used for signature verification. Ribeiro et al., (2011) experimented on the dataset GPDS-100 with an EER of 6.20% with the 12 genuine and 12 skilled forgery signatures.

*Deep Learning* is the trending research area. The model for offline signature recognition, which extracts high-level representation, was presented by Ribeiro et al., (2011). The high-level representation of signatures was extracted. The published result uses the other convenient features such as MDF, width, height, and conventional classifiers such as SVM. The Convolutional Neural Network verifies the system. CNN is used for feature extraction without the pre-knowledge of the data. The task is performed by the multilayer perceptron network (MLP). It is robust to signature location changes and scale variations. In this experiment, the database of 176 signatures is used from 22 subjects with a mean squared test error rate lower than 0.1%.

**2.1.5.** **Matching the template and query:** Matching the template and query by feature extraction or raw signature images. Abuhaiba, I. S. I., (2007), proposed that the signature verification depends only on the raw binary pixel intensities to avoid complex features. The method is justified as a graph matching problem for genuine and skilled forgery signatures produced by five subjects. The skilled forgery achieved an EER of 26.7%. Hiary et al., (2013) proposed the vertical projection features fed into a DTW algorithm for a stability factor. The system has the DER value of 22.5% on skilled forgery using a private database.

**2.1.6. Classification**: There are different classifiers for the offline signature verification system. The Bayes classifier is used by Ruiz-Del-Solar et al., (2008). The simplest Classifier used in this is KNN. The comparison between the two classifiers is made by Ribeiro et al., (2011) in KNN and PNN (Probabilistic neural networks). The genuine and forged signatures are divided into 12 genuine and 12 forged training and testing 42 samples. The best KNN result is 12.62% DER and the best CNN result is 12.33% on the gray level GPDS-160 database.

Neural networks are used by Xiao & Leedham, (1999) as two approaches. The first is with synthesis forgeries to train a neural network, and the other via feedback mechanism to recognize local stable parts by their node responses.

HMM, is a popular classifier used as a semi-automatic and combines a computer verification system with manual human verification. HMM, classifiers excel the individual human verifiers. As a result, HMM is the most efficient human classifier for operating costs.

**2.1.7. User-independent verification:** Eskander et al., (2013) studied the combination of writer-independent (W.I.) and writer-dependent (W.D.). By using a development database, the global classifier is designed. When the user joins the system, the verification is done by the W.I. classifier. During this, the samples are collected, and once adopted, the W.D. substitute in the place of the W.I. . In switching this verification system, the training samples are produced with higher accuracy by the W.D. classifier than the global W.I. classifier. The database used in this is GDPs-300, with 140 users as development set and 160 as training. The DER is 22.71% with the W.D. classifier, and with the W.I. classifier, the DER is 26.73%.

**2.1.8. User-Based score normalization**: For improving the accuracy in the biometric modalities, score normalization is used. Ferrer et al., (2005) studied the geometrical features which can be calculated by the fixed-point microprocessor such as smart-card. The robustness is tested by different classifiers such as HMM, SVM, and Euclidean distance.

**2.1.9. Models:** Parcham et al., (2021) proposed novel signature verification models to attempt modern problems. They used the combination of CNN and capsule neural networks for future feature extraction. They were using dual networks because it can reduce the number of layers and perimeters and decrease complexity. The capsule neural network detects the signatures, Pytel changes, and angular changes. Proposing this novel training mechanism reduced the number of layers in the SVM network. There was fast convergence and an increase in training capabilities. The datasets used in this paper were CEDAR, GPDS300, GPDS, synthetic signature, and BHsig260.

Pinzón-Arenas et al., (2019) propose a system that works on the architecture of DAG-CNN. The network consists of a CNN structure, configured layers, and a directed acyclic graph in which multiple parts and branches are there for finding the depth. We learn different features directly from the original network and verify the authenticity using the W.I. method. The data set consists of 339 genuine signatures and 260 forgery signatures. The preprocessing is done in two phases, first manually, and second, improving its performance and converting it into the grayscale and contrast adjustments of the signature.

Navid et al., (2019) present an experiment that constructed a CNN architecture using the Keras library. It is in the classification system for an image where each signature has a label. When any signature is entered into a program, it is compared with the other features of the different signatures with the same label. Compare features such as adjusting spacing and output probability that the signature is valid or not, check the image size. Vectors. After input, signatures are predefined. The validation rate is up to 25%.

Yapıcı, M. M., Tekerek, A., & Topaloğlu, N. (2021) proposed that the CNN-based validation system is used. The data is augmented by the cycle GAN method, which defines the image-to-image translation process. The hybrid system for data augmentation and classification better distinguishes authenticated and forged signatures. There are three phases involved; preprocessing, documentation and verification. They use four types of datasets; GPDS960 signature, 4NSigComp2010 Scenario 2, GPDS960 GRAY signature, MCYT75, and GPDS synthetic Signature. The proposed method achieved an FRR of 10.41%. FAR is equal to 8.66%, and E.R. equals 12.34%. And F1 score is equal to 88.97% for the GPDS synthetic signature database.

Rabbi et al., (2019) proposed that they used their data. Data were from 20 individuals. They trained it with different models CNN, CNN with data augmentation, 27 MLP, and SLP. Using a single-layer perceptron model, they obtained an accuracy of 39.9%. When they used the multilayer perceptron model, they obtained an accuracy of 63.57%. With the CNN model, they obtained an accuracy of 82.78%, which is higher than the last two. And when CNN with data argumentation was done the accuracy rate was 98.32%.

Jain et al., (2021) proposed that the CNN model with deep supervised learning architecture was used for classification. A trainable classifier and an automatic feature extractor are 2 parts of which CNN is composed. The two operations through which the feature extractor extracts features were convolution filtering and downsampling. Delta rule or gradient descent was the technique that lets you know the minimum value of the error function in weight space for the Backpropagation algorithm. The solution to the learning problem was considered by weights that minimize the error function. Libraries used are Keras, NumPy, pandas; Scikit-learn, matplot library, and scipy. The labeled folder has both forged and original English signatures.

## **2.2 Summary of Literature Review**

Table 1 Summary of Results of offline signature verifications

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Author** | **Dataset** | **Feature extraction** | **Classification technique** | **Results** |
| Rezaei, Mohammad  Naderi, Nader | 176 original Persian signatures | Number of feature maps and dimension of convolutional and subsampling filters | CNN | average of 99.86 has resulted in validation performance |
| Micheloni, C  Foresti, G L  Snidaro, L | MCYT database |  | hidden Markov modelling | 5% EER |
| Igarza, Juan J.  Hernaez, Inmaculada  Goirizelaia, Inaki  Espinosa, Koldo  Escolar, Jon | MCYT baseline Corpus | geometrical local features such as perimeter points for each blob | HMM models | LEN criterion never give EER values better than 30%, |
| Larkins, Robert  Mayo, Michael | CEDAR | graph matching and the Discrete Wavelet Transform | Manual and automatic classification | Automatic = 90.44%  Manual = 92.42 |
| Alonso-fernandez, F  Fairhurst, M C  Fierrez, J  Ortega-garcia, J | MCYT bimodal database | slant and envelope feature sub-vectors | global image analysis and statistical distance measures, andba |  |
| Francisco, Avda Madrid, Campus De Cantoblanco |  |  | - sed on local image analysis and HMM |  |
| Zulkarnain, Zuraidasahana  Mohd Rahim, Mohd Shafry  Othman, Nur Zuraifah Syazrah | GPDS-960 database | Aspect ratio, Pure height, Max. Horizontal histogram, Horizontal distance, Vertical distance, Hypotenuse distance  Normalized area of signature, Aspect ratio, Maximum histograms, Centroid, Trisurface, Sixfold surface, Number of edge points, Transition Maximum histogram and vertical histogram, Center of mass, Normalized area of signature, Aspect ratio, Tri surface feature, Sixfold surface feature, Transition feature  Depth, Vertical splitting, Horizontal  splitting  Geometric center, Vertical splitting, Horizontal splitting  Outer and inner contour, Slope of different strokes, Angle between two consecutive strokes Height, Width, Diagonal distance, Aspect ratio, Center of gravity, Area of the black pixel, Middle zone, Energy features | Euclidean distance  Neural Network (NN)  Euclidean distance  Neural Network  Mathematical Morphology  Correlation Technique | 84.1%  82.66%  79.2%  71.3%  58.0%  56.66% |
| C\u}celoğlu, \.Ilkhan  Oğul, Hasan | Tobacco-800 | Gradient-based features, Histogram of Oriented Gradients, Scale Invariant Feature Transform, Local Ternary Patterns, aspect ratio | Support Vector Machines (SVMs) |  |
| Pal, Srikanta  Pal, Umapada  Blumenstein, Michael | Hindi, and Bangla Signature Database, GPDS English Database(GPDS-160) | gradient features, contour point | SVM and MQDF | 83.60% |
| Nguyen, Vu  Blumenstein, Michael  Leedham, Graham | gpdsSIGNATURE | Modified Direction Feature, Maxima Feature, Ratio Feature | SVM | AER of 17.25% |
| Sabourin, Robert | database of 800 images consisting of 40 signatures written by 20 individuals | Characteristic Points and Mathematical Equations | minimum square method of curve |  |
| Hanmandlu, Madasu  Yusof, Mohd Hafizuddin Mohd  Madasu, Vamsi Krishna | 1200 handwritten signature images | Angle feature and distance feature | a fuzzy model such as the TS model | 85.4% |
| Gilperez, Almudena  Alonso-Fernandez, Fernando  Pecharroman, Susana  Fierrez, Julian Ortega-Garcia, Javier | MCYT database | Contour-Direction PDF, Contour-Hinge PDF, Direction Co-Occurrence PDFs, Run-Length PDFs | Normalization | EERs of 6.44% and 1.18% |
| Ruiz-Del-Solar, Javier  Devia, Christ  Loncomilla, Patricio  Concha, Felipe | GPDS signature database |  | Bayes classifier | FRR of 16.4% and a FAR of 14.2% |
| Eskander, George S.  Sabourin, Robert  Granger, Eric | GPDS-300  Brazilian database | directional probability density function | Writer- independent  Writer-Dependent | DER of 22.71% for WD and DER of 26.73 for WI |
| Ferrer, Miguel A.  Alonso, Jesús B.  Travieso, Carlos M. | Data from 160( 24 genuine and 30 forgeries) | Polar coordinates, cartesian coordinates | HMM, SVM and Euclidean-distance |  |
| Igarza, Juan J.  Hernaez, Inmaculada  Goirizelaia, Inaki  Espinosa, Koldo  Escolar, Jon | MCYT baseline Corpus | geometrical local features obtained | LR-HMM |  |
| Vargas, JF  Travieso, CM | 40 individuals(24 genuine only) | Stroke distribution in polar coordinates. | HMM and SVMLight model | 99.02% |
| Ribeiro, Bernardete  Gonçalves, Ivo Santos, Sérgio  Kovacec, Alexander Fukushima, Kunihiko | 300 individuals(24 genuine and 30 forgeries) | Image segmentation and image classification Pattern Recognition | Deep Learning Neural Network |  |
| Buhaiba, Ibrahim S.I.A | 5 subjects(15 genuine and 15 forgeries) | histograms of directional data or, horizontal and vertical projections | Neural Network and parallel processing | an equal error rate of 26.7% and 5.6% for skilled and random forgeries |
| Hiary, Hazem  Alomari, Raja  Kobbaey, Thaeer  Al-Khatib, Radi Z.  Al-Zu'Bi  Hasan, Hashem | Arabic handwritten signatures | Discrete wavelet transform, Reduction, | Image Registration, and Fusion | 6.23% |
| Xiao, Xu Hong  Leedham, Graham | 350 genuine signatures, 158 skilled forgeries, and 230 random forgeries | stroke distribution | Neural Network | FRR 6.7% on average |
| Ferrer, Miguel A.  Alonso, Jesús B.  Travieso, Carlos M. | 160 individuals( 24 genuine & 30 forgeries) | Outline Detection and Representation, Vector Based on Polar Coordinates, Vector Based on Cartesian Coordinates | HMM, SVM, and Euclidean Distance-Based Signature Model |  |
| Parcham, Ebrahim  Ilbeygi, Mahdi  Amini, Mohammad | CEDAR, GPDS300, GPDS, synthetic signature, and BHsig260 | Euclidean distance, Contrastive Loss function | CNN, Capsule neural network. | CBCapsNet: 92.94%, CNN-CapsNet: 88.1% |
| Pinzón-Arenas, Javier O.  Jiménez-Moreno, Robinson  Pachón-Suescún, César G. | 339 genuine signatures and 260 forgeries | location algorithm using morphological operations, | DAG-CNN | Validation accuracy : 99.4%, Subsequent testing: 99.3% |
| Navid, Shayekh Mohiuddin Ahmed  Priya, Shamima Haque  Khandakar, Nabiul Hoque  Ferdous, Zannatul  Haque, Akm Bahalul | SigComp 2011 | pattern recognition | CNN | 83% |
| Hameed, M. Muzaffar  Ahmad, Rodina  Kiah, Miss Laiha Mat  Murtaza, Ghulam | GPDS-Synthetic, GPDS-960, GPDS-300, GPDS-75 | Structural, statistical, texture, geometric and global transformation. | template matching models, traditional machine learning-based models, and neural network-based models |  |
| Yapıcı, Muhammed Mutlu  Tekerek, Adem  Topaloğlu, Nurettin | GPDS960 signature, 4NSigComp2010 Scenario 2, GPDS960 GRAY signature, MCYT75, and GPDS synthetic Signature |  | CNN models are VGG16, VGG19, ResNet50, and DenseNet121 | FRR:10.41%, FAR:8.66% |
| Rabbi, Md. Tariqulhasan Fazle  Rahman, S. M. Tanjilur  Biswash, Prokash  Kim, Jinsul  Sheikh, Alamin Saha, Aloke Kumar  Uddin, Mohammad Shorif | 20 individuals |  | CNN, CNN with data augmentation.MLP and SLP | SLP: 39.9%, MLP: 65.57%, CNN: 82.78%, CNN with augmentation: 98.32% |
| Jain, Kshitij Swapnil  Patel, Udit Amit  Kheni, Rushab | 2149 training images, 237 validation images, and 274 testing image | convolution filtering, downsampling | CNN | 99.21% |

## **2.3 Knowledge gap**

For a person to be accepted in any firm or an organization, a signature, especially a handwritten one, is considered the most legal way. Research in the field of signatures has progressed to such an extent that the researchers have incorporated various techniques from machine learning and pattern recognition to tackle the problem of recognition. Multiple factors have been used to get good results, for instance, designing new feature extractors and using them for other purposes. These results have been used in different fields, precisely computer vision, signal processing, and graphology.

Despite being so advanced in this particular field, to distinguish between reality and forgery, no more tremendous success has been achieved, as seen in the considerable error rates while performing the task when testing on a large dataset of people.

The methods for representation learning (feature learning) and checking how the problem is formulated by various techniques and the number of samples taken per person (genuine/forgery).

# **CHAPTER 3 METHODOLOGY**

This section describes the methodology proposed for recognizing isolated handwritten Punjabi signatures, numerals, and the half characters of each signature written using a ResNet50 approach. The technique uses a straightforward segmentation-based process, extracting features from individual signatures using a convolutional neural network as illustrated in (figure 8).

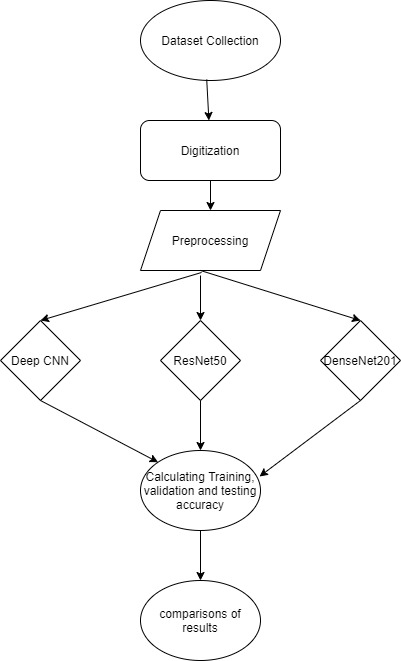


Figure 8 overall methodology

## **3.1 Dataset Collection**

To develop the contours-based offline signature verification system. In this phase, the first and foremost requirement is that of the dataset. Only after having the dataset then we proceed further with our topic. As specified in the objectives, we will build our dataset to address our problem. The first step in creating the dataset is the collection of the sample signatures from various writers. To collect signatures from the users, we provide some standard sheet designs. (Figure 9) shows the sheet structure we will be using for our research. We will be collecting five genuine signatures from each writer. Then we will collect ten simple forgeries for the same writer. Also, we will collect five skilled forgeries for the same writer. We collect the skilled forgeries by making the forger practice the original writer's signatures five times. Along with the signatures, we will also collect additional information about the Gender and Age of the original writer. After dataset collection, paper-based handwritten signature documents change into electronic format using a scanner by computers. The scanner scans a document and produces an electronic representation of image. Such an approach is generally used for offline recognition, where source images are obtained by scanning the printed, typewritten, handwritten documents or capturing a photograph through a digital camera.

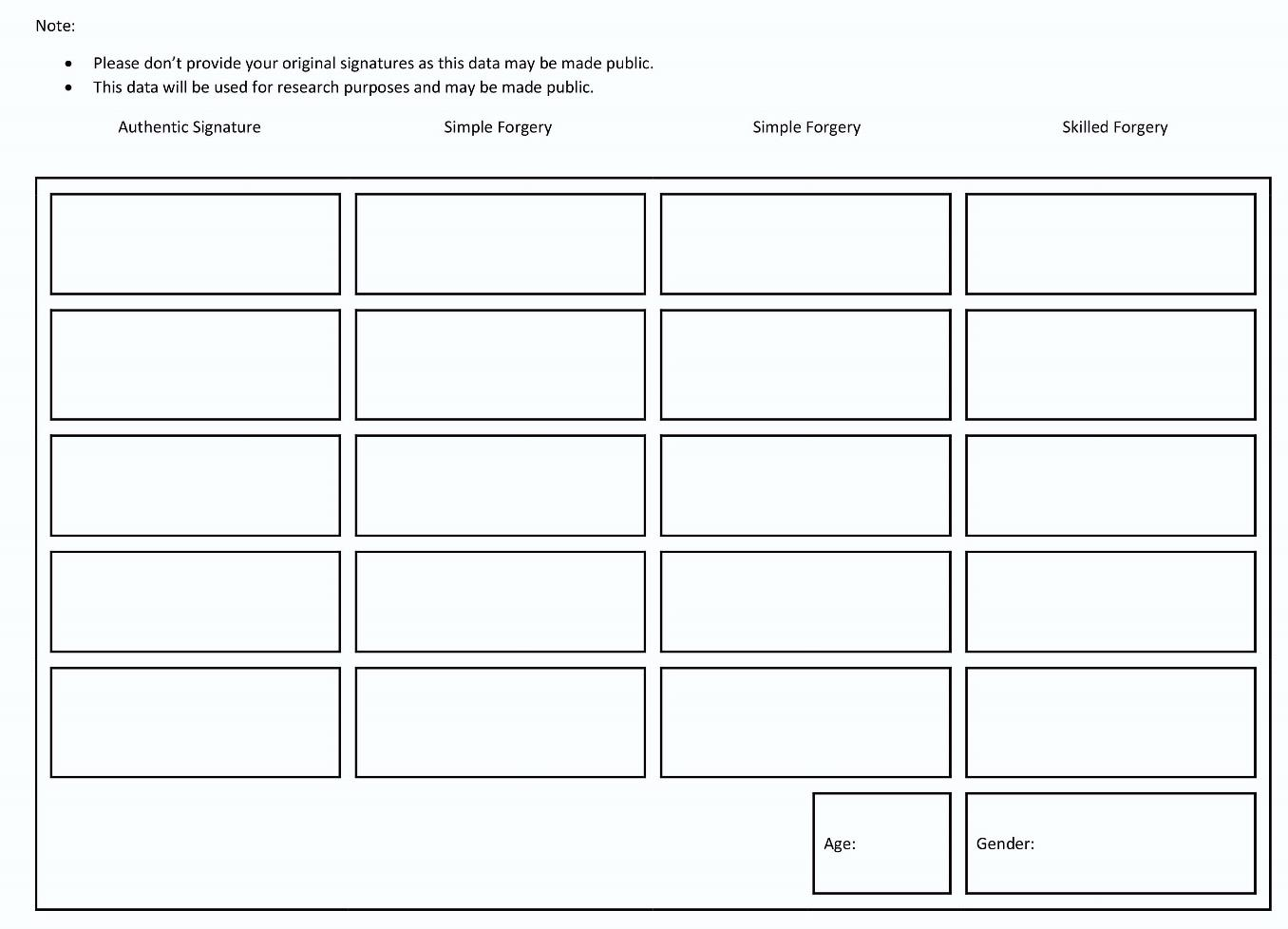


Figure 9 document design

We have collected 572 pages of data from various authors, each page consists of 20 signatures of which five signatures are authenticated and fifteen signatures are forged signatures as shown in figure 10 below. To get the distinguished form of each signature, each page was processed through different methods as discussed below. Each of the methods shows the corresponding input and particular output as results.

## 3.1.1 Digitization

This is the method of converting the handwritten page into an electronic format. A4- size paper is the input and figure 10 shows the output of that paper.

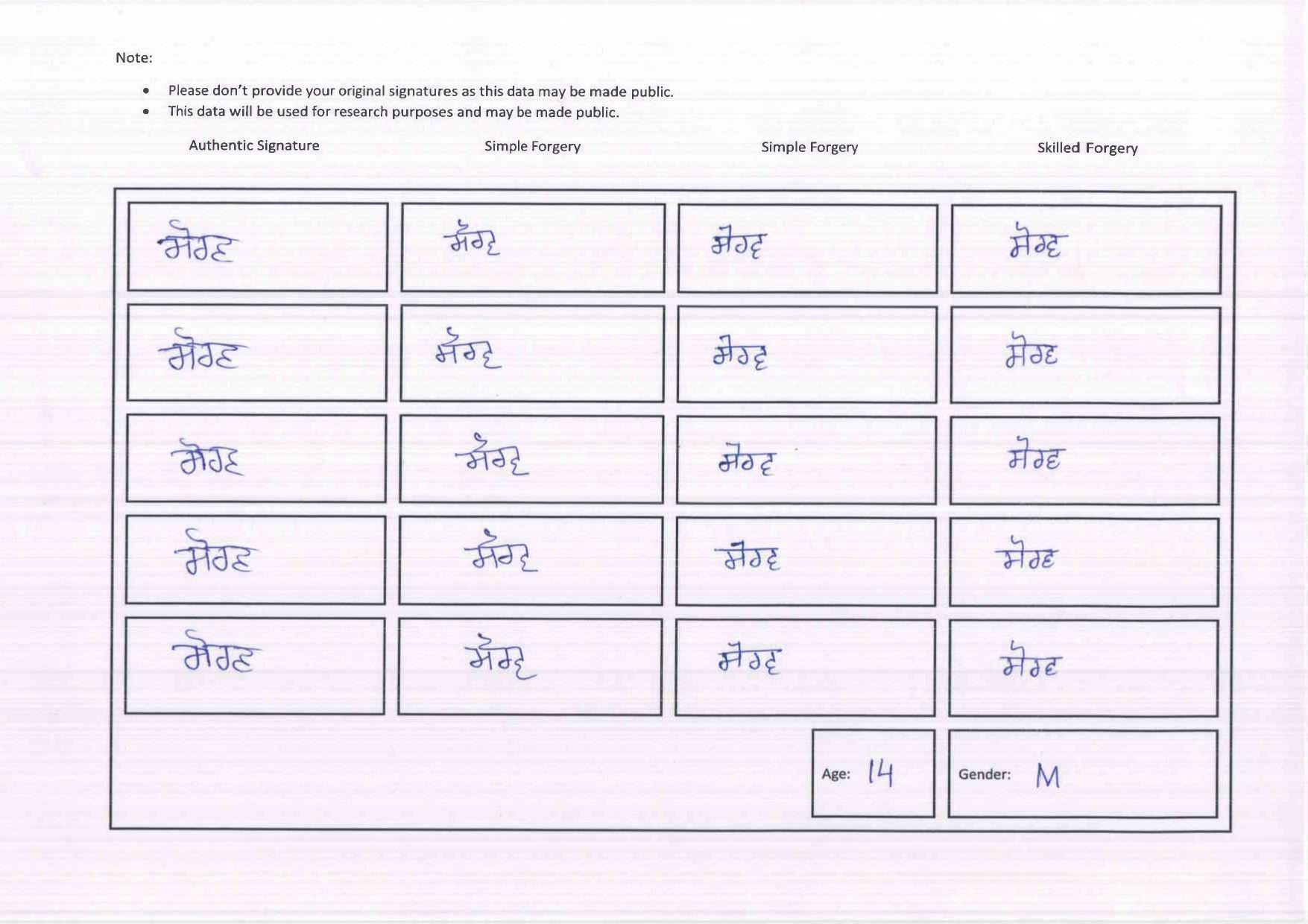
******

Figure 10 Handwritten document into electronic format

## 3.1.2 Box Detection and Cropping

Individual box of the paper which contains signatures have been detected, cropped, and stored in the extracted folder. The previous steps output is taken as input in this step and (Figure 11) will be the output.



Figure 11 Individual Signature

By performing the above methods we have collected data from the 572 authors. The description of the data presenting the total number of samples, genuine and forged are described below in the (table 2).

Table 2 Description of the datasets

|  |  |
| --- | --- |
| Description | Quantity |
| Number of authors | 572 |
| Number of samples | 11440 |
| Number of authenticated signatures | 2860 |
| Number of forged signatures | 8580 |
| Number of random forged signatures | 5720 |
| Number of skilled forged signatures | 2860 |

## **3.2 Preprocessing**

Preprocessing is an essential phase in automatic handwritten signature recognition. It involves a series of operations on the scanned input image to make it practical for later recognition phases and improve the overall performance. Should appropriately make image acquisition; otherwise, several issues like distortion, deformation, quality breakdown, orientation, and skewness may arise, introducing several problems and character. The proper preprocessing phase is significant for judging a successful contour-based engine. There are various preprocessing methods to remove image anomalies. These techniques involved image thresholding, noise removal, smoothing, discoing, thinning, image dilation, and normalization. The image processing steps are decsribed in the below table 3

Table 3 Pre-Processing Steps

### **3.2.1 Training Data and Testing Data**

After extracting the data from folders, the dataset is divided into 3 classes: training, testing and validation data. We uses the three models that are CNN 3D, DenseNet201 and ResNet50 for the verification and explains the details below:

#### **3.2.1.1 CNN**

The algorithm that we are using for our proposed counter-based system, basically CNN 3D, is the dimensional volume of neurons to classify handwritten signatures. The overall CNN algorithm, in which there are input layers and processed in such a way that the first convolutional layer having the zeropadding2D, batch normalization, each followed by a max-pooling layer and connected layers to perform classification. The input player is equivalent to the input image. The feature is then passed to another convolutional layer with many kernels to extract higher-level features of an input image. These help the CNN to learn a few features for smaller respective fields and more features for higher respective fields hence extracting more abstract features for the training. Each of the convolutional layers is then followed by the max pooling layer. To reduce the spatial dimensions of the output feature maps and reduce the number of learnable parameters.

After performing various operations, higher-level reasoning is performed with flattened and squashed as output features into dense layers. The proposed system connects all the neurons of the multiple layers to every single neuron. The number of classes are equal to the last fully connected layers for computing the class scores using softmax activation function. Convolutional and fully connected layer output, ReLU function activation function has been applied to increase the non-linearity of the decision function and the overall network without affecting their respective fields of convolutional layers. ReLU is more plausible to biological neurons, makes the trend in training significantly faster, and improves the generalization ability of deep neural network models.

There are several types of the CNN like AlexNet, VGG-16, Inception, LeNet, and DenseNet, ResNet50. Several types of CNNs varies in the internal structure filter sizes and layers, normal CNN took a lot of computational power and time.

**3.2.1.2 ResNet50**

ResNet is the residual networks which is the part of deep neural network. It is mainly used for the many computer visions such as object detection and image segmentation. There are total 50 layers deep. This network understand the many features representations for our datasets. This is much deeper than VGG16 and VGG19. In this basically the image is the multiple of 32 and 3 for height and width respectively. The model is used in the non-computer vision to reduce the computational expense also. The model architecture we are using is illustrated in (figure 12). The model that we are using in our system is shown the (figure 13)

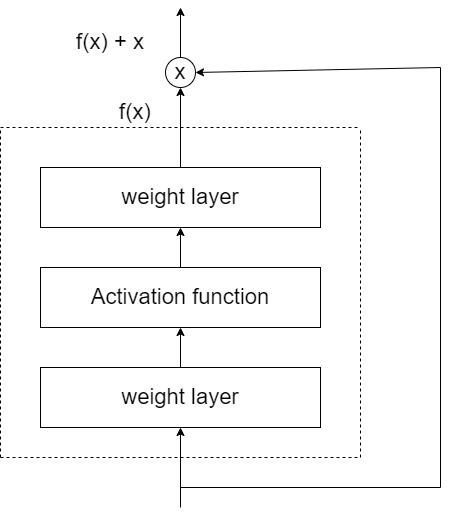


Figure 12 Res Net architecture

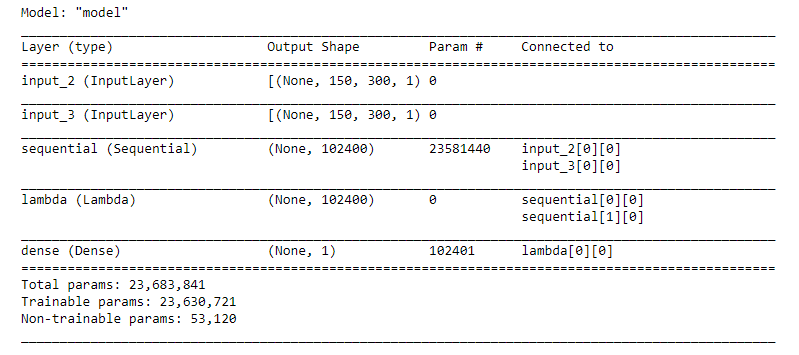


Figure 13 ResNet50 Model

**3.2.1.2 DenseNet201**

DenseNet201 is A 201 layers deep convolutional neural network. It is pre-trained version having millions of images from the ImageNet database. In these, each layer obtains additional input from all the preceding layers and passes on its own feature maps to all the subsequent layers, making the network thinner and compact. These specific features allow computational efficiency and memory efficiency. The model that we are using in our system is shown the (figure 14).

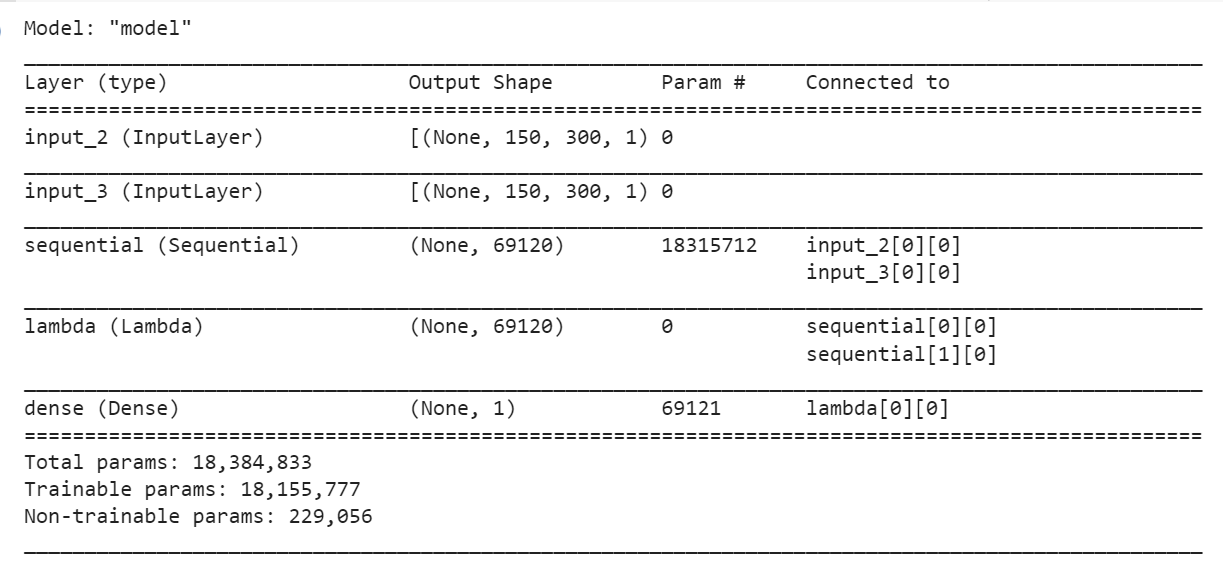


Figure 14 DenseNet201 Model

### **3.2.2 CNN Parameters**

*a. Learning Rate:* Learning Rate determines the speed at which the feedforward neural network is learning. The symbol α donates it. A small value leads to convergence, and a high value leads to divergence. An initial rate of 0.001% has been used in the proposed model. The formula used for the feedforward neural network is

y = f (x; θ)

b. *Adam optimizer:* This technique is basically used when we have large number of data samples or parameters because it requires less memory and efficient. It is basically a combination of ‘RMSP’ algorithm and ‘gradient descent with momentum’ algorithm.

*c. Mini-Batch:* Dividing the training data set into small subsets of instances is called a batch. The optimizer computes the gradient for each batch and updates the network parameters to minimize the error. This process is called iteration. The number of iterations to process, evaluate the gradient, and update the parameters of all batches of the training dataset is called an epoch. A model may require many such epochs until it learns the examples of the dataset. We have used a data size of 100 to train our model over 1 epochs in our model.

*d. Loss function:* It determines how well a specific algorithm models the given data. As a part of the Optimizer, the loss function repeatedly estimates the loss for the model's current state so that the network weights can be updated to reduce the failure in the subsequent evaluation. Our model used the categorical cross-entropy loss function for multiclass classification.

## **3.4 Performance Evaluation**

Following observations will be recorded in order to evaluate the results.

*Precision:* The precision basically tells the positive values in the model. It is defined as the number of true positives upon the sum of the number of true positives and false positives.

Precision = TP/TP+FP

*Recall:* it shows about the true predicted positive values in the model. It is defined as the number of true positive values upon the sum of the number of true positives values and the number of the false negative.

Recall = TP/TP+FN

*F1 score:* It is calculated as weighted average of precision and recall with taking both false positives and false negatives into account

F1 Score = 2\*(Recall \* Precision) / (Recall + Precision)

*Confusion matrix:* it is an N x N matrix used for performance evaluation of a classification model

Table 4 Confusion Matrix

|  |  |  |
| --- | --- | --- |
|  | Actual positive | Actual Negative |
| Predicted positive | TP | FP |
| Predicted negative | FN | TN |

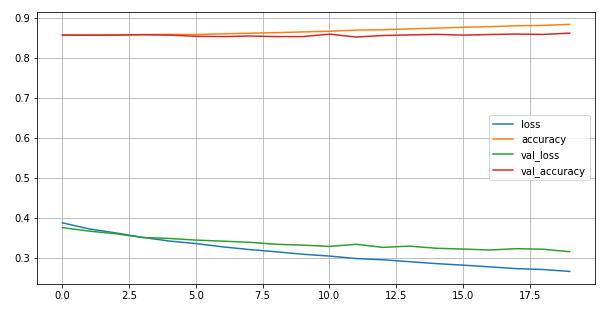
# **CHAPTER 4 RESULTS AND DISCUSSION**

This chapter describes the results obtained by our proposed system that we used and discussed earlier. In our system, there are a total of 11440 samples in which there are 2860 authenticated signatures and 8580 forged signatures. ResNet50 performs well on the writer-independent model as in the writer-independent signature verification model.

Table 5 Losses and accuracies of different models

|  |  |  |  |
| --- | --- | --- | --- |
| Results | ResNet50 | DenseNet201 | CNN |
| No. of epochs | 20 | 20 | 10 |
| Training accuracy | 0.8841 | 0.8568 | 0.2556 |
| Training Loss | 0.2656 | 0.4078 | 11.3522 |
| Validation accuracy | 0.8622 | 0.8164 | 0.2772 |
| Validation Loss | 0.3150 | 0.4231 | 1.1951 |
| Testing accuracy | 0.87309 | 0.5340 | 0.6731 |
| Testing Loss | 0.1269 | 1.3517 | 5.855 |

## **4.1 Results obtained by ResNet50**

****

Accuracy

Parameters

Figure 15 Results obtained by Resnet50

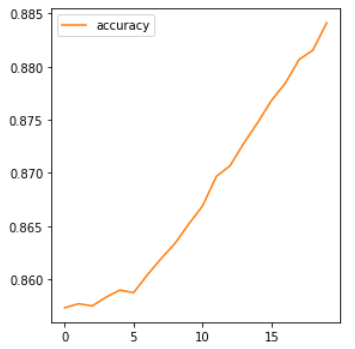
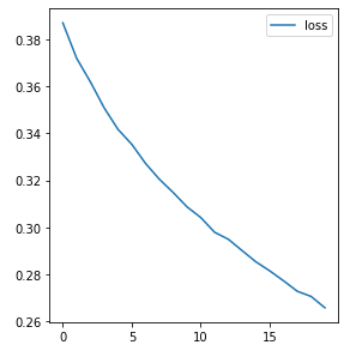
** **

Figure 16 Accuracy Graph Figure 17 Loss in Model

The precision, recall and F1 score that we calculated via implementing the ResNet50 is shown in table 6 and confusion matrix is shown in the (figure 18).

Table 6 Precision, recall and F1 score of ResNet50

|  |  |
| --- | --- |
| Precision | 0.8758 |
| Recall | 0.8804 |
| F1 score | 0.8248 |
| Threshold | 2 |

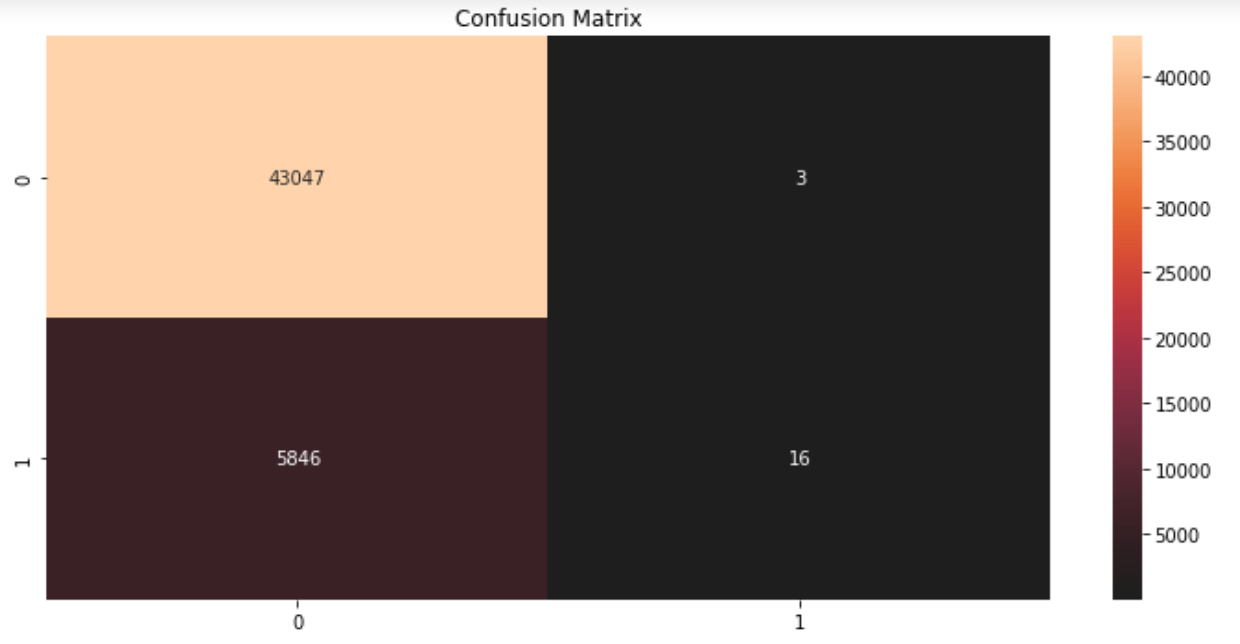


Figure 18 Confusion Matrix of ResNet50

## **4.2 Results Obtained by the DenseNet201**

The precision, recall and F1 score that we calculated via implementing the ResNet50 is shown in table 7. The results obtained by the DenseNet201 is shown in (figure19) and confusion matrix is shown in the (figure 20).

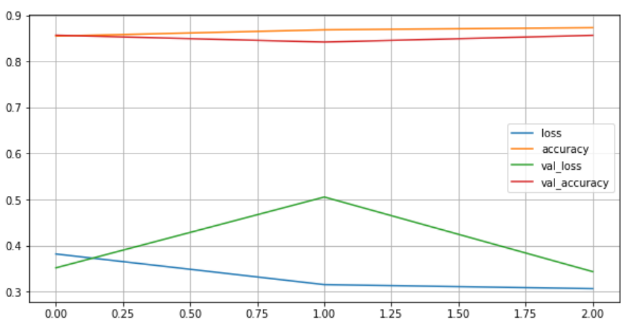
****

Figure 19 Results obtained by the DenseNet201

Table 7 Precision, recall and F1 score of DenseNet201

|  |  |
| --- | --- |
| Precision | 0.7975 |
| Recall | 0.5340 |
| F1 score | 0.6146 |
| Threshold | 2 |

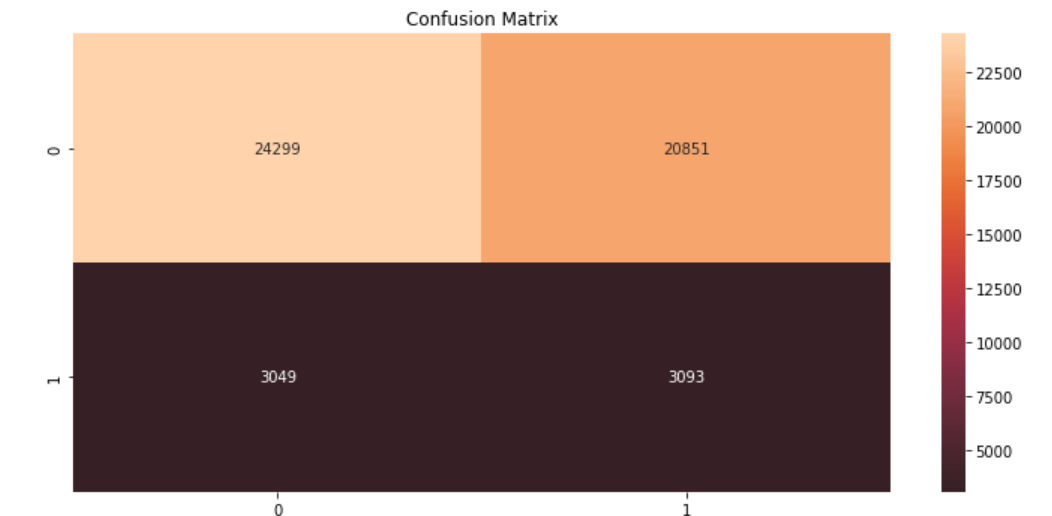


Figure 20 Confusion Matrix of DenseNet201

## **4.3 Results Obtained by the CNN**

The precision, recall and F1 score that we calculated via implementing the ResNet50 is shown in table 8. The ROC curve is shown in the figure 21.

Table 8 Precision, recall and F1 score of CNN

|  |  |
| --- | --- |
| Precision | 0.50 |
| Recall | 0.3749 |
| F1 score | 0.4286 |
| Threshold | 0.5850 |

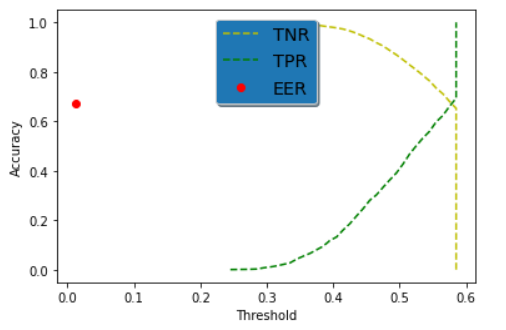


Figure 21 ROC Curve by the CNN

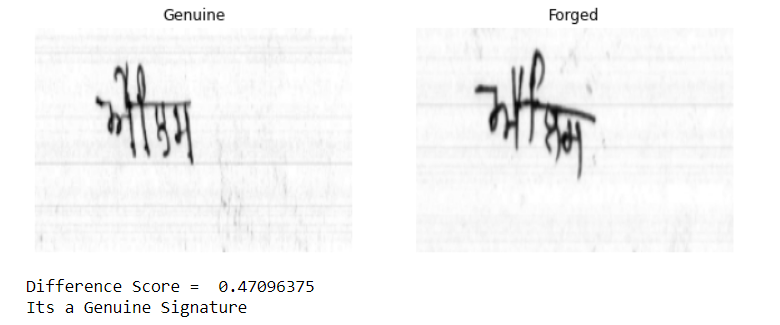


Figure 22 Difference Scores In the images

As the above (figure 22) shows the genuine and forged signature and as we expected there are lots of difference so there will be high difference score but as we checked there are only 0.4709 difference score it is representing on we can’t rely on this model for the verification.

## **4.4 Comparison with existing systems**

The recognition rate reached up to 88% is the best result obtained by ResNet50 technique proposed for the Punjabi language to the best of our knowledge because of the complex nature of the Punjabi language. The different accuracies of the different researches are given below in the table 9.

Table 9 Comparisons with the existing techniques

|  |  |  |
| --- | --- | --- |
| Authors(Year) | Technique used | Highest Recognition Rate |
| Rabbi et al., (2019) | CNN, CNN with data augmentation, MLP, and SLP | SLP: 39.9%, MLP: 65.57%, CNN: 82.78%, CNN with augmentation: 98.32% |
| Jain et al., (2021) | CNN | 99.21% |
| Navid et al., (2019) | CNN | 83% |
| Pinzón-Arenas et al., (2019) | DAG-CNN | Validation accuracy: 99.4%, Subsequent testing: 99.3% |
| Yapıcı, M. M., Tekerek, A., & Topaloğlu, N. (2021) | CNN-based validation system | FRR = 10.41%.  FAR = 8.66% |
| Parcham et al., (2021) | CNN, Capsule neural network | CBCapsNet: 92.94%, CNN-CapsNet: 88.1% |

# **CHAPTER 5 CONCLUSION AND FUTURE SCOPE**

## 

## **5.1 Conclusion**

In this research work, we proposed a new dataset for handwritten signatures named Handwritten Punjabi signature Dataset (HPSD). The data is gathered from 572 peoples from the Bathinda region from various schools and. The basic motive of making the dataset is only to provide the dataset publicly so that in future there will be more research. There are various challenges in the offline Punjabi signature due to their complexity in writing pattern. The challenge is resolved by the CNN model architecture ResNet50 which obtains an accuracy of 87.08%. As we observed, increase in depth of convolutional layer, the image verification model enhance the performance. As ResNet50 reduce the overfitting and remove the gradient problem.

## **Future work**

In future, we intend to implement the model in which we select more new features so that we can get more accurate results for the various complex languages like Hindi, Bengali, and Farsi. Design the model for the highly rotated signatures so that it can adapt it easily. In the security domain also, increase the security for the model that it can implement in the real life problems. For the model develop the mobile and web application.

# **REFERENCES**

Alonso-Fernandez, F., Fairhurst, M. C., Fierrez, J., & Ortega-Garcia, J. (2007). Automatic measures for predicting performance in off-line signature. *Proceedings - International Conference on Image Processing, ICIP*, *1*. https://doi.org/10.1109/ICIP.2007.4378968

Cordeiro Bastos, L., Bortolozzi, F., Sabourin, R., & Kaestner, C. (1997). *Mathematical Modelation of Handwritten Signatures by Conics*.

Coetzer, J., Herbst, B. M., & du Preez, J. A. (2004). Offline signature verification using the discrete radon transform and a hidden Markov model. *Eurasip Journal on Applied Signal Processing*, *2004*(4), 559–571. https://doi.org/10.1155/S1110865704309042

# Eskander, G. S., Sabourin, R., & Granger, E. (2013). Hybrid writer-independent writer-dependent offline signature verification system. *IET Biometrics*, *2*(4), 169–181. <https://doi.org/10.1049/iet-bmt.2013.0024>

Ferrer, M. A., Alonso, J. B., & Travieso, C. M. (2005). Offline geometric parameters for automatic signature verification using fixed-point arithmetic. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *27*(6), 993–997. <https://doi.org/10.1109/TPAMI.2005.125>

Gilperez, A., Alonso-Fernandez, F., Pecharroman, S., Fierrez, J., & Ortega-Garcia, J. (2008). Offline Signature Verification Using Contour Features. *11th International Conference on Frontiers in Handwriting Recognition, Montréal, Québec - Canada, August 19-21, 2008*. <http://atvs.ii.uam.es>

Hanmandlu, M., Yusof, M. H. M., & Madasu, V. K. (2005). Offline signature verification and forgery detection using fuzzy modeling. *Pattern Recognition*, *38*(3), 341–356. <https://doi.org/10.1016/j.patcog.2004.05.015>

Hiary, H., Alomari, R., Kobbaey, T., Al-Khatib, R. Z., Al-Zu'Bi, & Hasan, H. (2013). Offline signature verification system based on DWT and common features extraction. *Journal of Theoretical and Applied Information Technology*, *51*(2), 165–174.

Igarza, J. J., Hernaez, I., Goirizelaia, I., Espinosa, K., & Escolar, J. (2005). Offline signature recognition based on dynamic methods. *Biometric Technology for Human Identification II*, *5779*, 336. <https://doi.org/10.1117/12.603557>

Larkins, R., & Mayo, M. (2008). Adaptive feature thresholding for offline signature verification. *2008 23rd International Conference Image and Vision Computing New Zealand, IVCNZ*, *October 2014*. <https://doi.org/10.1109/IVCNZ.2008.4762072>

Micheloni, C., Foresti, G. L., & Snidaro, L. (2003). INTELLIGENT DISTRIBUTED SURVEILLANCE SYSTEMS A network of co-operative cameras for visual surveillance. *Image (Rochester, N.Y.)*, 205–212. <https://doi.org/10.1049/ip-vis>

Nguyen, V., Blumenstein, M., & Leedham, G. (2009). Global features for the offline signature verification problem. *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, *September 2014*, 1300–1304. https://doi.org/10.1109/ICDAR.2009.123

Pal, S., Pal, U., & Blumenstein, M. (2013). Multi-script offline Signature verification: A two stage approach. *CEUR Workshop Proceedings*, *1022*(January 2013), 31–35.

Ribeiro, B., Gonçalves, I., Santos, S., & Kovacec, A. (2011). Deep learning networks for offline handwritten signature recognition. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *7042 LNCS*, 523–532. <https://doi.org/10.1007/978-3-642-25085-9_62>

Ruiz-Del-Solar, J., Loncomilla, P., & Devia, C. (2008). Fingerprint verification using local interest points and descriptors. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *5197 LNCS*, 519–526. <https://doi.org/10.1007/978-3-540-85920-8_63>

Sabourin, R. (1997). Offline signature verification by local granulometric size distributions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *19*(9), 976–988. <https://doi.org/10.1109/34.615447>

Xiao, X. H., & Leedham, G. (1999). Signature verification by neural networks with selective attention. *Applied Intelligence*, *11*(2), *213–223*. <https://doi.org/10.1023/A:1008380515294>

Jain, K. S., Patel, U. A., & Kheni, R. (2021). Handwritten Signatures Forgery Detection. *International Research Journal of Engineering and Technology*, *1989–2003*. [www.irjet.net](http://www.irjet.net)

Navid, S. M. A., Priya, S. H., Khandakar, N. H., Ferdous, Z., & Haque, A. B. (2019). Signature Verification Using Convolutional Neural Network. *2019 IEEE International Conference on Robotics, Automation, Artificial-Intelligence and Internet-of-Things, RAAICON 2019*, 35–39. <https://doi.org/10.1109/RAAICON48939.2019.19>

Parcham, E., Ilbeygi, M., & Amini, M. (2021). CBCapsNet: A novel writer-independent offline signature verification model using a CNN-based architecture and capsule neural networks. *Expert Systems with Applications*, *185*(April), 115649. <https://doi.org/10.1016/j.eswa.2021.115649>

Pinzón-Arenas, J. O., Jiménez-Moreno, R., & Pachón-Suescún, C. G. (2019). Offline signature verification using DAG-CNN. *International Journal of Electrical and Computer Engineering*, *9*(4), 3314–3322. <https://doi.org/10.11591/ijece.v9i4.pp3314-3322>

Rabbi, M. T. F., Rahman, S. M. T., Biswash, P., Kim, J., Sheikh, A., Saha, A. K., & Uddin, M. S. (2019). Handwritten Signature Verification Using CNN with Data Augmentation. *The Journal of Contents Computing*, *1*(1), 25–37. <https://doi.org/10.9728/jcc.2019.12.1.1.25>

Yapıcı, M. M., Tekerek, A., & Topaloğlu, N. (2021). Deep learning-based data augmentation method and signature verification system for offline handwritten signature. Pattern Analysis and Applications, 24(1), 165–179. <https://doi.org/10.1007/s10044-020-00912-6>

Hameed, M. M., Ahmad, R., Kiah, M. L. M., & Murtaza, G. (2021). Machine learning-based offline signature verification systems: A systematic review. Signal Processing: Image Communication, 93(October 2020), 116139. <https://doi.org/10.1016/j.image.2021.116139>

*Punjabi grammar - Jatland Wiki*. (n.d.). Retrieved April 27, 2022, from <https://www.jatland.com/home/Punjabi_grammar>

*Punjabi Language - Jatland Wiki*. (n.d.). Retrieved April 27, 2022, from <https://www.jatland.com/home/Punjabi_Language>

Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. In *Journal of Big Data* (Vol. 8, Issue 1). Springer International Publishing. <https://doi.org/10.1186/s40537-021-00444-8>

Patil, A., & Rane, M. (2021). Convolutional Neural Networks: An Overview and Its Applications in Pattern Recognition. *Smart Innovation, Systems and Technologies*, *195*, 21–30. <https://doi.org/10.1007/978-981-15-7078-0_3>

Singh, P., Verma, P., & Singh, N. (2021). *Offline Signature Verification: An Application of GLCM Features in Machine Learning*. *Annals of Data Science*, *0123456789*. <https://doi.org/10.1007/s40745-021-00343-y>

Souza, V. L. F., Oliveira, A. L. I., & Sabourin, R. (2018). A writer-independent approach for offline signature verification using deep convolutional neural networks features. *Proceedings - 2018 Brazilian Conference on Intelligent Systems, BRACIS 2018*, *December*, 212–217. <https://doi.org/10.1109/BRACIS.2018.00044>

SR, D. (2012). Offline Signature Verification Using Neural Network. *International Journal of Machine Intelligence*, *4*(1), 408. <https://doi.org/10.9735/0975-2927.4.1.408-408>

Engin, D., Kantarci, A., Arslan, S., & Ekenel, H. K. (2020). Offline signature verification on real-world documents. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, *2020*-*June*, 3518–3526. <https://doi.org/10.1109/CVPRW50498.2020.00412>

Merrouch, R. (2021). *Sources included in the report*. 1–37.

Mushtaq, F., Misgar, M. M., Kumar, M., & Khurana, S. S. (2021). *UrduDeepNet: offline handwritten Urdu character recognition using deep neural network*. Neural Computing and Applications, 0123456789, 15229–15252. https://doi.org/10.1007/s00521-021-06144-