International Journal of Advanced Engineering **ISSN: 2457-0397**

Vol. 2, Issue I, Jan-Mar, 2019 http://bharatpublication.com/current-issue.php?jID=29/IJAE

LEVERAGING THE TENSOR FLOW IN THE PREDICTION AND TRANSLATION OF HANDWRITTEN GURMUKHI DIGITS

Mukul Ganghas

ABSTRACT

Savvy Character Recognition or Optical Character Recognition has been a prominent place of studies in the field of PC vision. The acknowledgement of manually written character or digit is a hard venture due to range within the composing style of an individual; one aspect can compose a similar e-book in various manners by making a touch variety in protecting a pen. In the past, many neural community classifiers or fashions had been formed with the aid of taking an exclusive array of nodes in the hidden layer, epochs, and gaining knowledge of charge with masses of complexity like lengthy codes, more reminiscence usage and so on. Python and its libraries have made the complicated implementation clean and compact. Keras is a python library and is an excessive stage neural community API that runs on the pinnacle of Tensorflow or Theano (different Python libraries). In this paper, we proposed a secure neural community version of the use of Keras to understand remoted Gurmukhi digit and compare the version of the use of confusion metrics. Dataset turned into amassed from a hundred ninety extraordinary people, preprocessed and represented in binary form to present entry to the classifier. Consequently, we have carried out a maximum accuracy of ninety-eight—forty-one %.

I. INTRODUCTION

Intelligent Character Recognition (ICR) is a method wherein computer acknowledges handwritten characters of digitized file mechanically. It is one of the most entrancing and testing zones, of example, acknowledgement with distinctive right down to earth utility opportunities. The development of an automation method is truly putting off the difference between guy and gadget in many packages. No one may be puzzled if a few devices are pleasing the mission in half of the time as compared to manual processing of that mission. Various sensible applications of the ICR system are - (1) Vehicle Number Plate popularity, (2) Reading aid for the blind, (three) Speech to the text entered into the laptop for computing device booklet, online transactions, ledgering, and so on. (4) Sorting of postal mail, financial institution cheques (5) Document transcription and (6) Natural Language Processing, and so on. Like an Optical Character Recognition (OCR) device, the ICR system has additionally undergone six stages (1) Digitization, (2) Preprocessing, (three) Segmentation, (four) Feature Extraction, and (5) Classification and (6) Recognition or Prediction. Every degree has its own importance. Digitization approach to transforming the bodily record into virtual form in order that laptop can process it. In image processing, this digitization degree refers to the technique of sampling and quantization. Further two levels, Preprocessing and Segmentation together is known as Low feature extraction step. In low feature extraction steps, it's far required to nice preprocessor tune the enter snapshots like do away with noise, conversion from RGB or grey photo to binary photograph for quicker processing and if more than one characters are written collectively on a single sheet, segment into a person for proper recognition. The fourth degree is Feature Extraction, it's also referred to as High degree function extraction step, in this situation whole awareness can be on the primary pixel values or interest points of an image which wonderful every person from different characters [1]. It is difficult to manually fetch all the pixel values of a photograph which helped in character formation. There are diverse strategies that are used to bring the pixels of interest from a picture and used the fetched (pixel values) functions as enter to numerous system mastering classifiers like KNN(K-Nearest Neighbors) or SVM(Support Vector Machine). With Artificial neural community as the classifier, it isn't always required to feed the classifier manually with the high stage features due to the fact artificial neural network imitates the human mind intelligence through learning the ones features itself using returned propagation and gradient descent method. The performance of this kind of system has usually been significantly dependent on the selection of appropriate amounts and preprocessing of the dataset. If the model is educated over a massive quantity of preprocessed dataset, the version can be sturdy in performance and spotting the man or woman.

II. SCRIPT AND DATASET

Gurmukhi is a script that includes the techniques to put in writing the Punjabi language. The Punjabi language is the nearby language of Punjab state of India, which is located in northern a part of us of a. In Canada, the Punjabi language is taken into consideration as the fifth widely spoken language after English, French, Mandarin, and Cantonese. Local language has a prime effect on the country's work. All most essential obligations like voter identification shape, riding license shape, etc. are available in the nearby language of the precise kingdom for the convenience of rural areas of a nation. The individual and numeric set of Punjabi language is as shown under

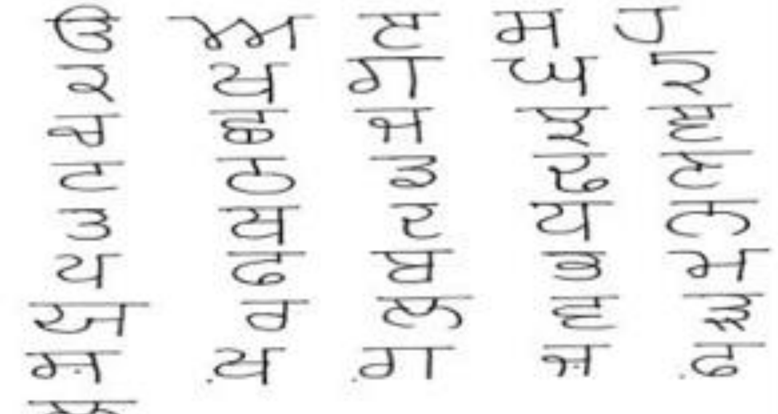


Fig1. Character set of Punjabi Language

Fig2: Numeric Set of Punjabi Language

The dataset for the Handwritten Gurmukhi Digit has been amassed on an A4 size white sheet with an everlasting marker with nib length 0.8. The dataset is collected from a hundred ninety particular age institution human beings like College Students, Prophets operating in spiritual places, humans running in the financial institution, and so on. Each man or woman has written 70 samples of each 0-9 Punjabi digits.

III. PROPOSED RECOGNITION SYSTEM

The proposed Handwritten Gurmukhi digit recognition using a simple neural network classifier consists of the following stages [2]. Each Stage has its role and importance.

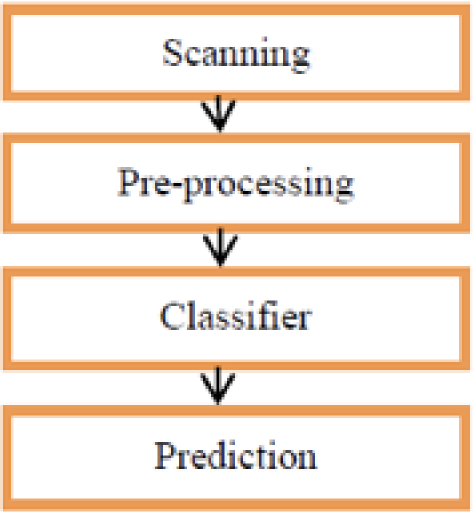


Fig3. Image Recognition Steps using ANN

Scanning
Handwritten Digit samples were acquired on a plain white sheet. These samples have been scanned the use of a flat-bed scanner with 300 dpi and saved the scanned samples within the image format, i.E. Jpg shape.

Pre-processing
Pre-processing transforms the records into the format intending to be higher without difficulty and efficaciously processed. It is the crucial and preliminary step due to the fact 70 digits are altogether written on a single A4 size sheet, want to crop each finger, and perform resize and denoising. The following are the stairs that might be used to perform the preprocessing.

· Raw dataset is first scanned and transformed .Jpg grey-scale snapshots.

· The rectangle is formed. The use of contour around each digit and right boundary of the strokes is decided, and the redundant heritage region is cropped.

· The grey-scale cropped picture is first inverted and makes it a binary photograph of the use of the Adaptive threshold binary.

· Finally, the image is resized to a well-known size of 32 x 32 with the centre of the contour as the centre of the resized photo.

Classifier

Simple Neural Network classifier is which includes fully linked layers handiest or additionally called Dense layers in Keras. That method, all the nodes in a single layer are connected to the nodes inside the other sheet. A dense class of Keras is used to train the later secure neural network version for our dataset. Compact elegance, in particular, has four arguments to skip i.e., several nodes in layer, an amount of entering values, weight or kernel initializer for the network, and last activation function used. Our simple neural community is along with three primary segments – Input Layer, hidden layer, and output layer. The amount of hubs in the data layer is equivalent to the scale of the image; for example, the tallness and width of the dim scale picture. For our scenario, the picture size is 32 x 32, so there will be 784 hubs inside the records layer.

Furthermore, there might be ten hubs in the yield layer as we have (0-nine) class names. The amount of centres within the concealed layer is a hyperparameter; by way of and big, it is usual that if it's worth ought to be saved in the middle of information layer hubs, and yield layer hubs at that point outcomes are agreeable. As it's miles a heuristic method, so the procedure of trial and error can lead us to the beautiful shape of the network. The weight for the community can be initialized the use of Keras routinely, one handiest want to skip the value as "ordinary" for small random range generator. ReLU (Rectified Linear Unit) activation function is used to get the output of each layer and bypass it to the opposite layer. In the last segment, the output layer, we will use the softmax feature because of the activation characteristic because the softmax feature is used to calculate the probability of all ten elegance labels. The one with the most possibility could be the expected magnificence label. During the compiling of the model, the Logarithmic Loss function, which is likewise called categorical go entropy characteristic in Keras, is used to calculate the loss, optimizer function i.E. Adam (Adaptive moment estimation) is used to replace the load at some point in the schooling of the community.

Training of network commonly means to discover the ideal quality weights which provide us maximum Accuracy of the model. The hyperparameters are want to set in suit function of Keras, like the number of epochs, batch size (these once more can be determined using trial and error technique). Evaluate the overall version performance using the compare function with a unique dataset that isn't always used for education in the model.

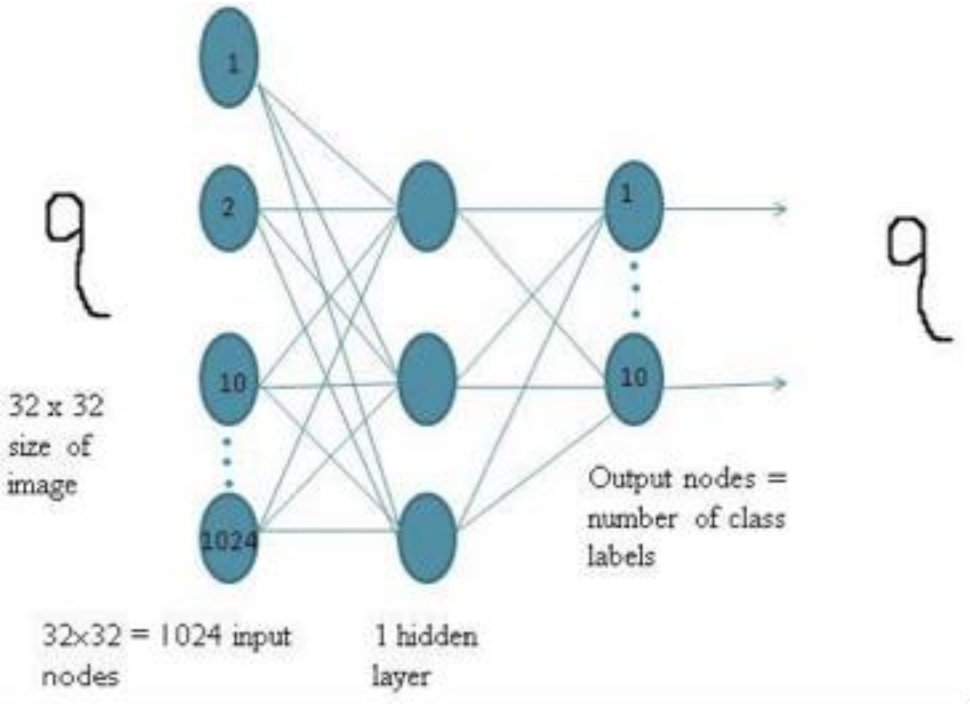


Fig 4. Neural network architecture for digit image classification

Prediction
When the favoured accuracy effects carried out from a classifier, you possibly can shop the model the use of h5 format. HDF5py is a python package that helps binary information format and stores a large number of numerical statistics, and one could manage that statistics the use of NumPy. The saved version can be accessed through the use of the predicted characteristic to predict the Gurmukhi digit picture. Predict feature can display its NumPy label that is kept in the model in addition to the corresponding class result.

IV. EXPERIMENT AND RESULT DISCUSSION

To implement the dataset, the use of python, we used the Pycharm development device, and Keras is a high-level neural network API that is used to create the neural network and deep learning fashions. It ensures quick experimentations. Keras uses either Tensorflow or Theano as backend, but in our case, we have used the TensorFlow as backend. Tensorflow plays the mathematical operations on arrays within the backend. Other libraries that we used are OpenCV, NumPy, matplotlib, pillow. The undertaking turned into a run on a sixty-four-bit operating device, Intel centre i5 (7th technology) CPU @2.50GHz 2.70 GHz, with eight GB RAM. Following is the flow chart of making the neural network classifier and predicting the digit the usage of Keras.

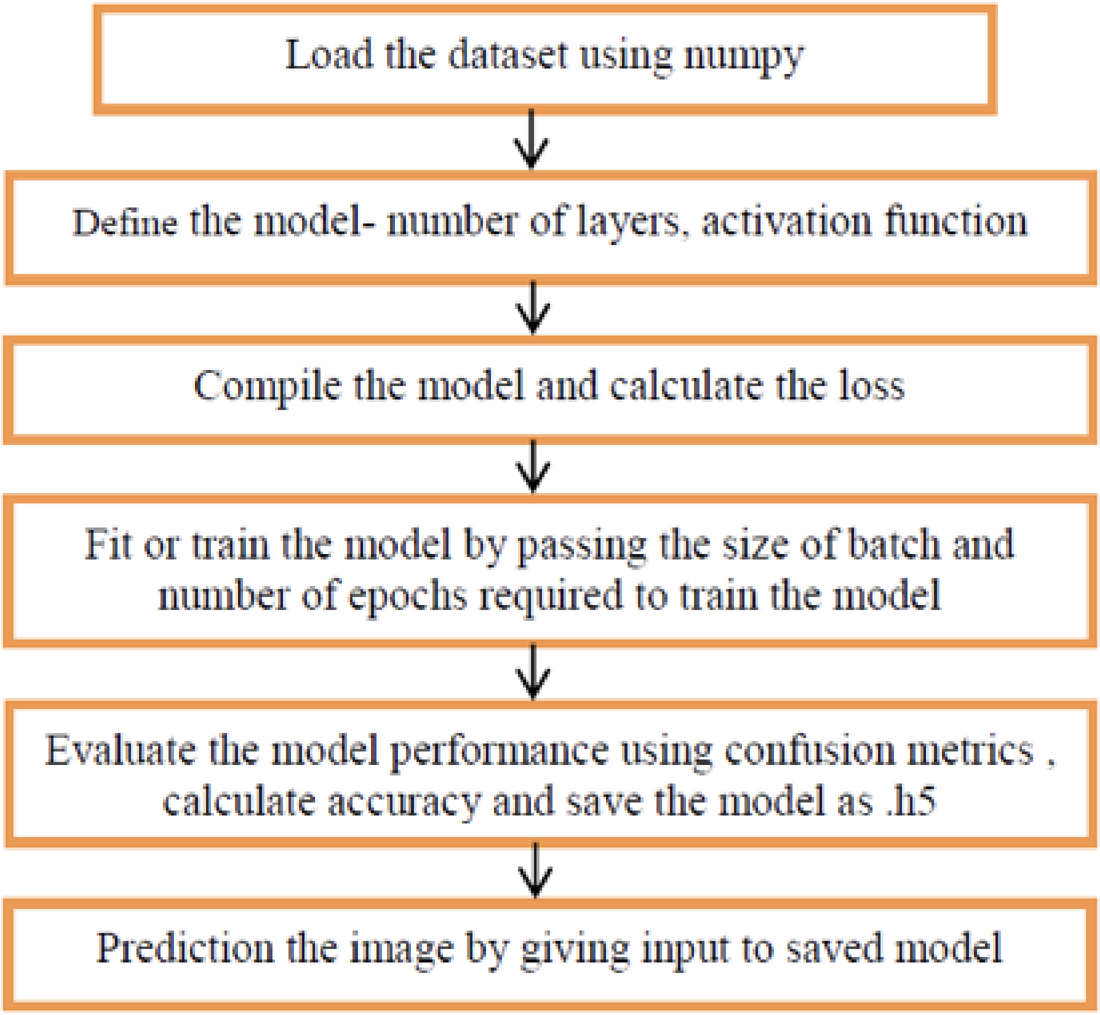
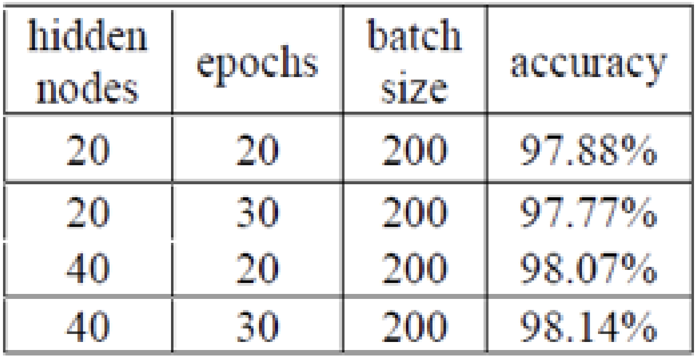

Load the dataset using numpy
Define the model- number of layers, activation function
Compile the model and calculate the loss
Fit or train the model by passing the size of batch and number of epochs required to train the model
Evaluate the model performance using confusion metrics , calculate accuracy and save the model as .h5
Prediction the image by giving input to saved model

Fig 4. Flowchart of creating classifier using Keras

The version turned into trained on 10559 pix and validate on 2640 photographs. The result table is proven under wherein extraordinary values of hyperparameters have been set and record the Accuracy. The version parameters which gave the most Accuracy, that model has been saved with.H5 extension, and also, it's miles used for the prediction cause.

TABLE1 ACCURACY RECORD WITH DIFFERENT HYPERPARAMETERS



hidden nodes	epochs	batch size	accuracy
20	20	200	97.88%
20	30	200	97.77%
40	20	200	98.07%
40	30	200	98.14%

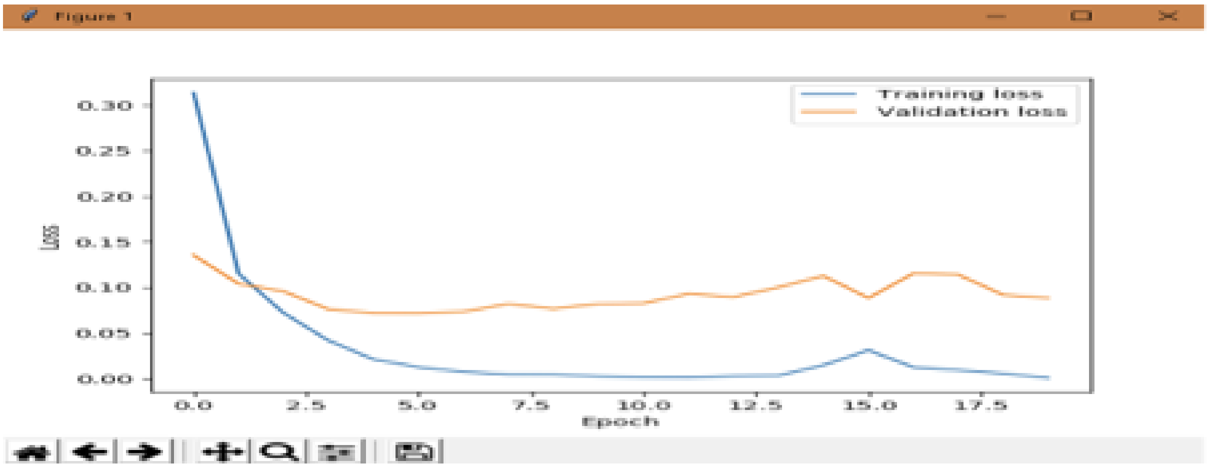
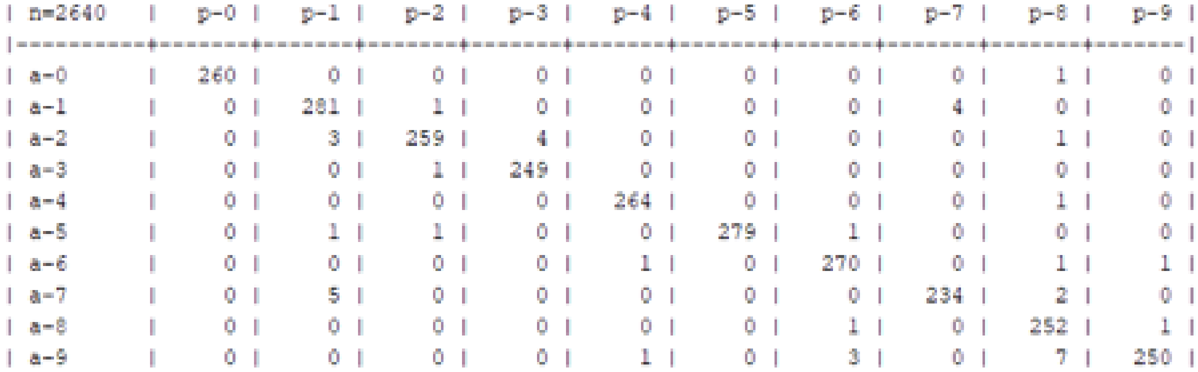


Fig5 Resultant Graph of Epoch vs Loss

TABLE2. EVALUATION OF MODEL USING CONFUSION MATRIX



n=2640	p-0	p-1	p-2	p-3	p-4	p-5	p-6	p-7	p-8	p-9
a-0	260	0	0	0	0	0	0	0	1	0
a-1	0	281	1	0	0	0	0	4	0	0
a-2	0	3	259	4	0	0	0	0	1	0
a-3	0	0	1	249	0	0	0	0	0	0
a-4	0	0	0	0	264	0	0	0	1	0
a-5	0	1	1	0	0	279	1	0	0	0
a-6	0	0	0	0	1	0	270	0	1	1
a-7	0	5	0	0	0	0	0	234	2	0
a-8	0	0	0	0	0	0	1	0	252	1
a-9	0	0	0	0	1	0	3	0	7	250

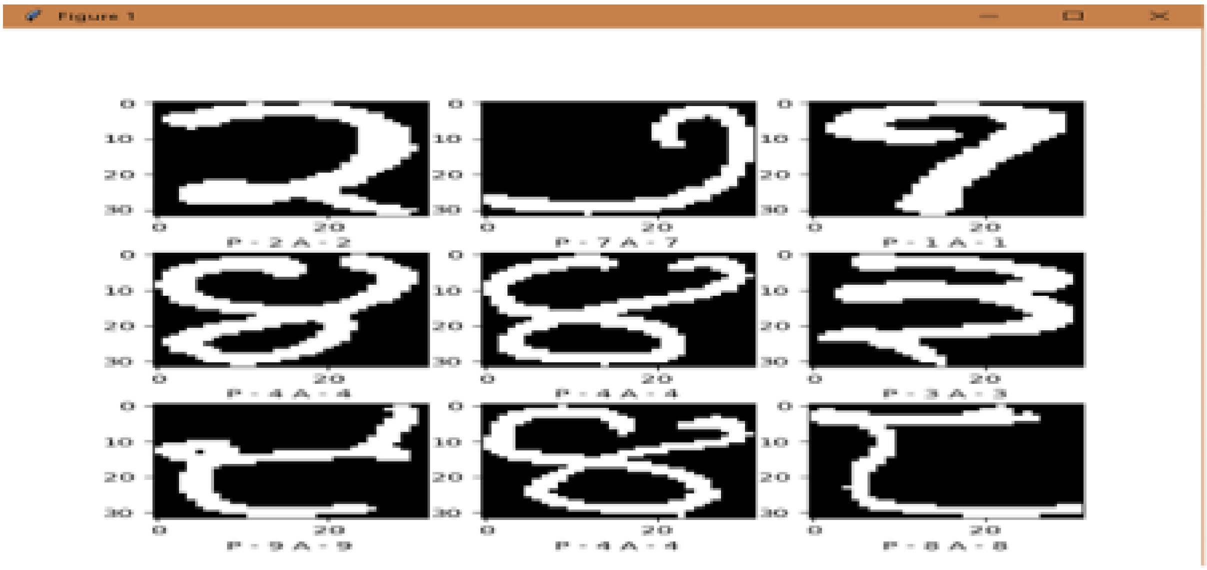


Fig 6 Random nine Number predicted by the model

Table1. Shows that the performance of the community depends on the hyperparameter, and the high cost of a hyperparameter may be located with an empirical system only. A small variety of nodes in the hidden layer will give much less Accuracy; however, if we're educating the version with a more excellent

range of epochs, Accuracy may also boom as much as a few factors. Still, then it saturates at a certain point, and the massive number of nodes in the hidden layer will ultimately increase the Accuracy depending on the number of epochs and batch size. We can examine this type of circumstance with baby, if a toddler is gaining knowledge of 1 to10 number continuously for ten days, it's going to have correctly taken hold of over numbers, and he can supply the right result while checking out. But if it's miles mastering 1 to 100 numbers consistently for ten days, it can not deliver Accuracy as accurate as it had given while studying 1 to 10 digits. Because we increase the batch length i.E. Numbers amount, but epochs i.E. Variety of days had been the same. If a child has extreme grabbing power, he may even provide better Accuracy; this is a new variety of nodes inside the hidden layer. So Accuracy indeed receives affected with the trade in the values of the hyperparameters. Table 2. Shows the bring about the form of misunderstanding matrix over the take a look at records which depicts that which numbers have been successfully expected and which numbers were wrongly anticipated via the version. Fig 6 suggests the graph between loss and epoch for both education and validation dataset, which states that we stored the model which had given us the highest Accuracy and record the consequences of prediction and also shown the confusion matrix.

V. CONCLUSION

In this paper, we've got implemented the handwritten Gurmukhi digit recognition using a secure neural community with Keras and TensorFlow. Keras is an excellent python library for one to apprehend the running of a neural network as it has many built-in training and features that can directly be used and made the running less difficult. One doesn't need to write down long codes. Python has rendered the coding an easy project for a learner. Though it has given common effects towards spotting digits. This Accuracy can be stepped forward either via the usage of more number of hidden layers or through using deep neural network algorithms for pattern reputation. Intelligent Character Recognition proposed in this paper carried out on isolated Gurmukhi digits. It can similarly be extended to multi-digit, multi-person, or paragraph status on intended noisy sheets.