# Artificial Intelligence Documentation

**Dataset link:** [**https://www.kaggle.com/datasets/crawford/emnist**](https://www.kaggle.com/datasets/crawford/emnist)

## **Optical Character Recognition (OCR)**:

Optical Character Recognition (OCR) is the process that converts an image of text into a machine-readable text format. For example, if you scan a form or a receipt, your computer saves the scan as an image file. You cannot use a text editor to edit, search, or count the words in the image file. However, you can use OCR to convert the image into a text document with its contents stored as text data.



# **OCR IMPORTANCE:**

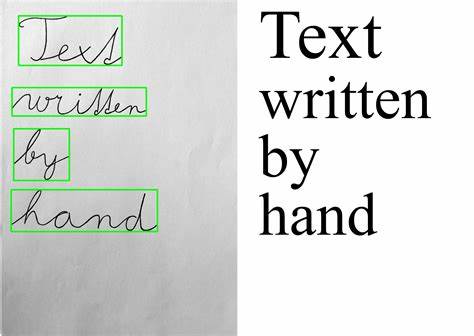
Most business workflows involve receiving information from print media. Paper forms, invoices, scanned legal documents, and printed contracts are all part of business processes. These large volumes of paperwork take a lot of time and space to store and manage. Though paperless document management is the way to go, scanning the document into an image creates challenges. The process requires manual intervention and can be tedious and slow.

Moreover, digitizing this document content creates image files with the text hidden within it. Text in images cannot be processed by word processing software in the same way as text documents. OCR technology solves the problem by converting text images into text data that can be analyzed by other business software. You can then use the data to conduct analytics, streamline operations, automate processes, and improve productivity.

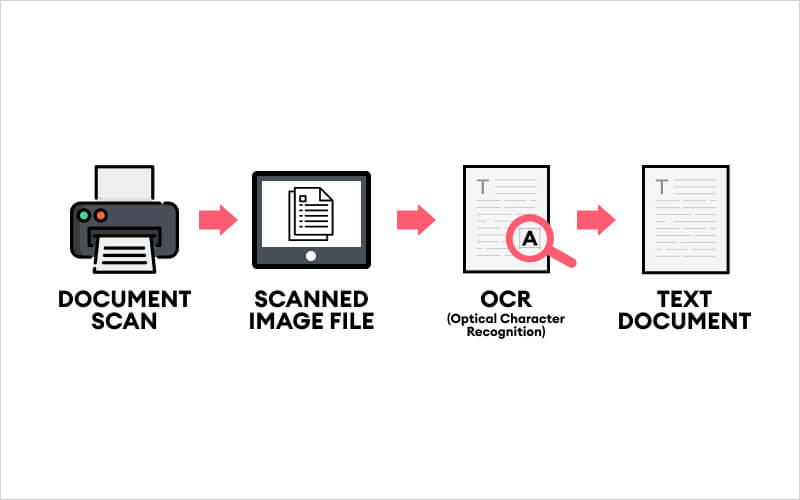
## **What is handwriting recognition?**

Handwriting recognition, also known as handwriting OCR or cursive OCR, is a subfield of [OCR technology](https://research.aimultiple.com/ocr-technology) that translates handwritten letters to corresponding digital text or commands in real-time. To perform this task, these systems benefit from pattern matching to identify various styles of handwritten letters. Wikipedia defines handwriting recognition as:

“the ability of a computer to receive and interpret intelligible handwritten input from sources such as paper documents, photographs, touch-screens, and other devices.”



The Process of OCR:  
In the following, we will show how optical character recognition works and explain the main steps of traditional OCR technologies.



1. Scanning the Document:

This is the prime step of OCR, which connects to a scanner to scan the document. Scanning the document decreases the number of variables to account for when creating the OCR software since it standardizes the inputs. Also, this step specifically enhances the efficiency of the entire process by ensuring perfect alignment and sizing of the specific document. This initial step can also include object detection, to focus subsequent vision processing tasks on specific image areas.

1. Refining the Image

In this step, the optical character recognition software improves the elements of the document that need to be captured. Any imperfections, such as dust particles, are eliminated, and edges, as well as pixels, are smoothed to get a plain and clear text. This step makes it easier for the program to capture data while being able to clearly “see” the words being inputted without, for instance, smudges or irregular dark areas. Such image processing tasks are essential in all types of vision pipelines, to sharpen or auto-brighten images. OpenCV provides a toolset that is often used for such tasks

1. Binarization

The refined image document is then converted into a bi-level document image, containing only black and white colors, where black or dark areas are identified as characters. At the same time, white or light areas are identified as background. This step aims to apply segmentation to the document to easily differentiate the foreground text from the background, which allows for the optimal recognition of characters.

1. Recognizing the Characters

In this step, the black areas are further processed to identify letters or digits. Usually, an OCR focuses on one character or block of text at a time. The recognition of characters is carried out by using one of the following two types of algorithms:

* Pattern recognition. The pattern recognition algorithm involves inserting text in different fonts and formats into the OCR software. The modified software is then used for comparing and recognizing the characters in the scanned document.
* Feature detection. Through the feature detection algorithm, OCR software applies rules considering the features of a certain letter or number to identify characters in the scanned document. Examples of features include the number of angled lines, crossed lines, or curves used for comparing and identifying characters. Such text recognition techniques are the basis of most deep learning OCR methods.

Simple OCR software compares the pixels of every scanned letter with an existing database to identify the closest match. However, sophisticated forms of OCR divide every character into its components, such as curves and corners, to compare and match physical feature with corresponding letters.



OCR Applications:

**Data Entry Automation:**

OCR can automate the process of data entry by extracting information from paper documents and converting it into machine-readable text. This reduces manual data entry efforts and minimizes errors.

**Invoice Processing:**

Businesses use OCR to extract relevant information from invoices, such as vendor details, amounts, and dates. This accelerates the processing of invoices and improves accuracy.

**Form Recognition:**

OCR is employed to recognize and extract data from forms, surveys, and questionnaires. This speeds up the process of information collection and analysis.

**Banking and Finance:**

OCR is utilized in the banking sector for tasks like reading checks, extracting information from financial documents, and processing various forms.

**Text Search in Images:**

OCR enables the searching of text within images. This is useful for image indexing, content retrieval, and searching through scanned documents.

**Passport and ID Verification:**

OCR is employed in identity verification processes, such as reading information from passports, driver's licenses, and other identification documents.

**Healthcare Records:**

OCR is used to convert handwritten or printed medical records into digital formats, making it easier for healthcare professionals to access and manage patient information.

**Automated Receipt Scanning:**

OCR helps in scanning and extracting data from receipts, facilitating expense tracking, and simplifying the reimbursement process for businesses.

**Text-to-Speech Conversion:**

OCR can be combined with text-to-speech technology to assist visually impaired individuals by converting printed or handwritten text into spoken words.

**Translation Services:**

OCR is integrated into translation applications to convert printed text from one language to another. This is useful for quickly translating documents or signs.

**Postal Services:**

OCR is used in postal services for sorting and processing mail by automatically reading addresses and routing information.

**Legal Document Processing:**

Law firms and legal departments use OCR to convert physical legal documents into searchable digital formats, making it easier to retrieve and analyze information.

**Education:**

OCR is applied in educational settings for grading exams, converting printed material into electronic formats, and facilitating accessibility for students with visual impairments.

**Mobile OCR Apps:**

There are numerous mobile applications that use OCR for tasks like scanning business cards, translating text, and extracting information from images.

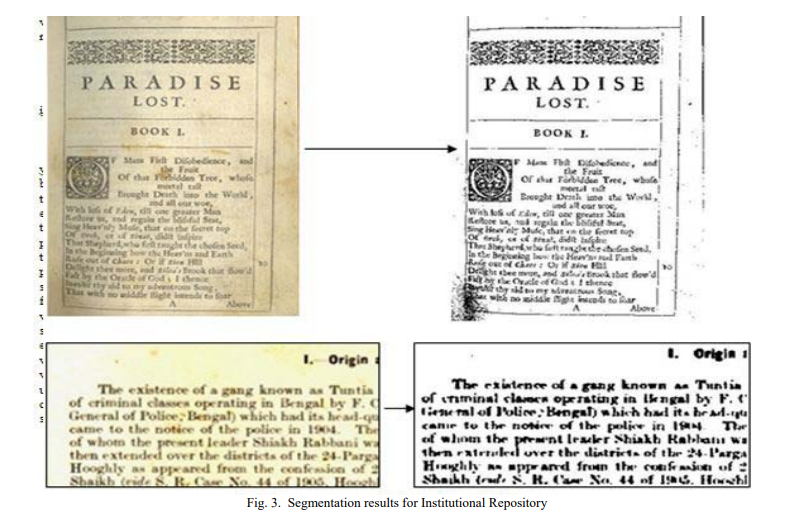
# microsoft azure cognitive service for vision

Academic publications(papers):

1. Retrieving OCR Text: A Survey of Current Approaches  
   Description:  
   The importance of effectively retrieving OCR text has grown significantly in recent years,  
   surveyed current techniques in use to facilitate information retrieval on collections of OCR text. There are many proposed techniques and models available for use, and hopefully in the future a generalized solution that takes on aspects of all available techniques will be available to members of the Information Retrieval community

Link: http://www.ijmlc.org/papers/137-L0022.pdf

1. Survey of OCR Applications  
   Description:  
   The paper presents a survey of applications of OCR in different fields and further presents the experimentation for three important applications such as Captcha, Institutional Repository and Optical Music Character Recognition. We make use of an enhanced image segmentation algorithm based on histogram equalization using genetic algorithms for optical character recognition. The paper will act as a good literature survey for researchers starting to work in the field of optical character recognition  
   link: http://www.ijmlc.org/papers/137-L0022.pdf



1. OPTICAL CHARACTER RECOGNITION TECHNIQUE ALGORITHMS

Description:

In this paper, we present a new neural network (NN) based method for optical character recognition (OCR) as well as handwritten character recognition (HCR). Experimental results show that our proposed method achieves increased accuracy in optical character recognition as well as handwritten character recognition. We present through an overview of existing handwritten character recognition techniques. All the algorithms describes more or less on their own. Handwritten character recognition is a very popular and computationally expensive task; we describe advanced approaches for handwritten character recognition  
link: <https://www.jatit.org/volumes/Vol83No2/15Vol83No2.pdf>

A diagram of a network

Description automatically generated

1. A Survey on Optical Character Recognition System

Description:

Optical Character Recognition (OCR) has been a topic of interest for many years. It is defined as the process of digitizing a document image into its constituent characters. Despite decades of intense research, developing OCR with capabilities comparable to that of humans remains an open challenge. Due to this challenging nature, researchers from industry and academic circles have directed their attention towards Optical Character Recognition. Over the last few years, the number of academic laboratories and companies involved in research on Character Recognition has increased dramatically

Link: <https://arxiv.org/ftp/arxiv/papers/1710/1710.05703.pdf>

A diagram of a character recognition system

Description automatically generated

1. REVIEW ON OPTICAL CHARACTER RECOGNITION

Description:

In the research works revised in this paper, character recognition system use different approaches and many of them get good accuracy. What we can understand from this paper is feature extraction techniques should be choose according to the character you working because each scripts or alphabets has its own nature therefor need to find techniques which fit or suitable for characters. The better able to extract features from character more we can detect and recognize characters in highest accuracy.

Link: <https://www.researchgate.net/profile/Muna-Ahmed/publication/334162853_REVIEW_ON_OPTICAL_CHARACTER_RECOGNITION/links/5d1af333a6fdcc2462b74595/REVIEW-ON-OPTICAL-CHARACTER-RECOGNITION.pdf>

A diagram of a model

Description automatically generated

## EMNIST

The EMNIST dataset is a set of handwritten character digits derived from the NIST Special Database 19 and converted to a 28x28 pixel image format and dataset structure that directly matches the MNIST dataset.

## Format

There are six different splits provided in this dataset and each are provided in two formats:

1. Binary (see emnist\_source\_files.zip)
2. CSV (combined labels and images)
   * Each row is a separate image
   * 785 columns
   * First column = class label (see mappings.txt for class label definitions)
   * Each column after represents one pixel value (784 total for a 28 x 28 image)

A table with numbers and a number on it

Description automatically generated

Our dataset is only one split which is EMNIST-letters:

### A black square with white letters and text Description automatically generated

### Letters datasets

The EMNIST Letters dataset merges a balanced set of the uppercase and lowercase letters into a single 26-class task.

* train: 88,800
* test: 14,800
* total: 103,600
* classes: 37 (balanced)

C. EMNIST Letters Results:

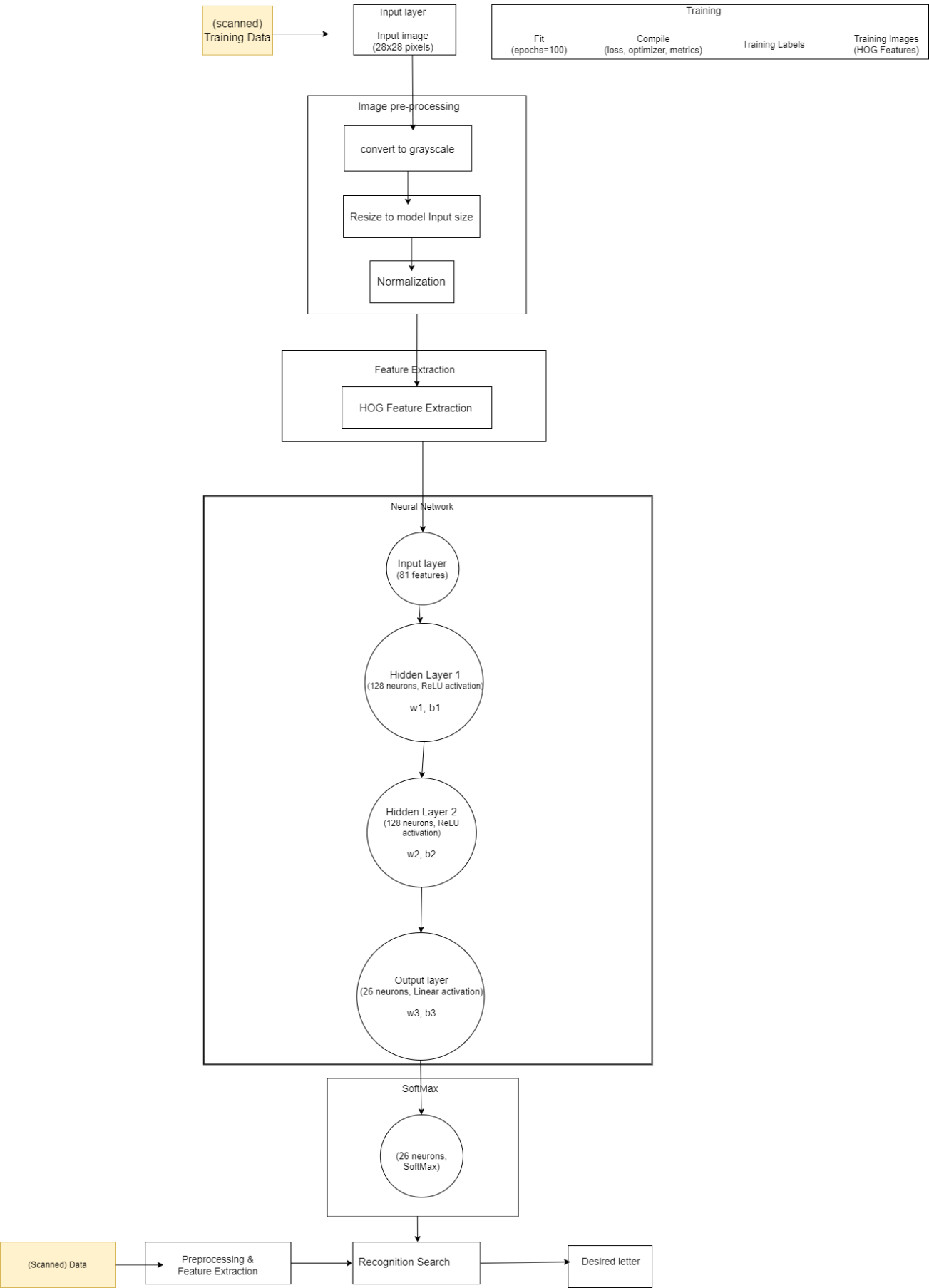
The EMNIST Letters dataset was created to mitigate the issues regarding case and the misclassification of letters and digits that plague the By Class and By Merge datasets. This is accomplished by combining the uppercase and lowercase versions of each letter into a single class and removing the digit classes entirely. This dataset therefore provides a different classification task from the other datasets and poses a letteronly classification task in the spirit of the original MNIST dataset. As certain letters have distinctly different uppercase and lowercase representations, the classifiers are required to associate two different representations of a letter with a single class label

The EMNIST Letters dataset seeks to further reduce the errors occurring from case confusion by merging all the uppercase and lowercase classes to form a balanced 26-class classification task

A screenshot of a document

Description automatically generated

Diagrams:

Block diagram:  


Flowchart

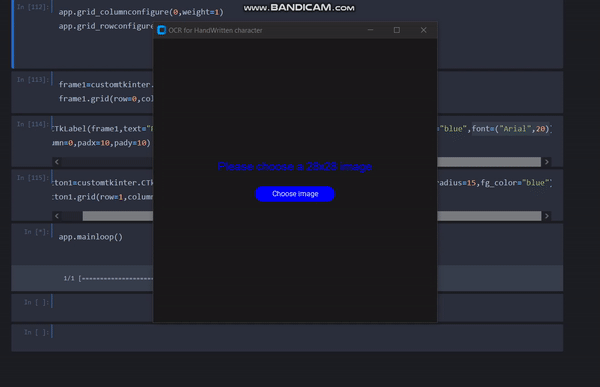
A diagram of a process

Description automatically generated

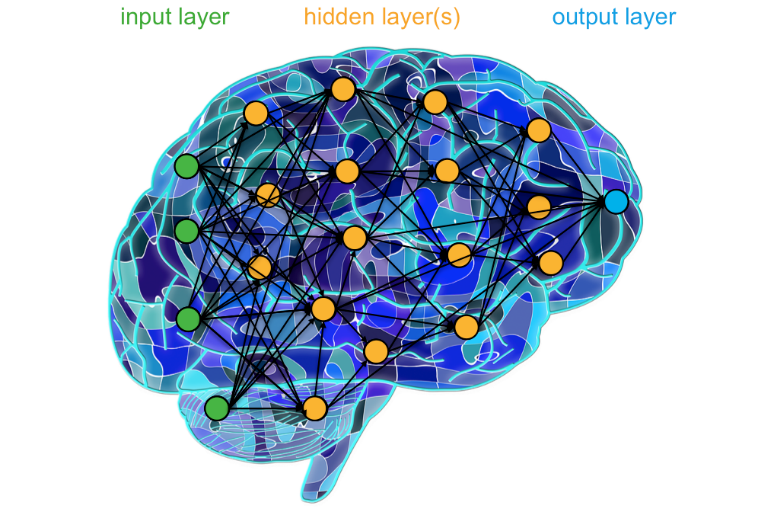
Graphical User Interface

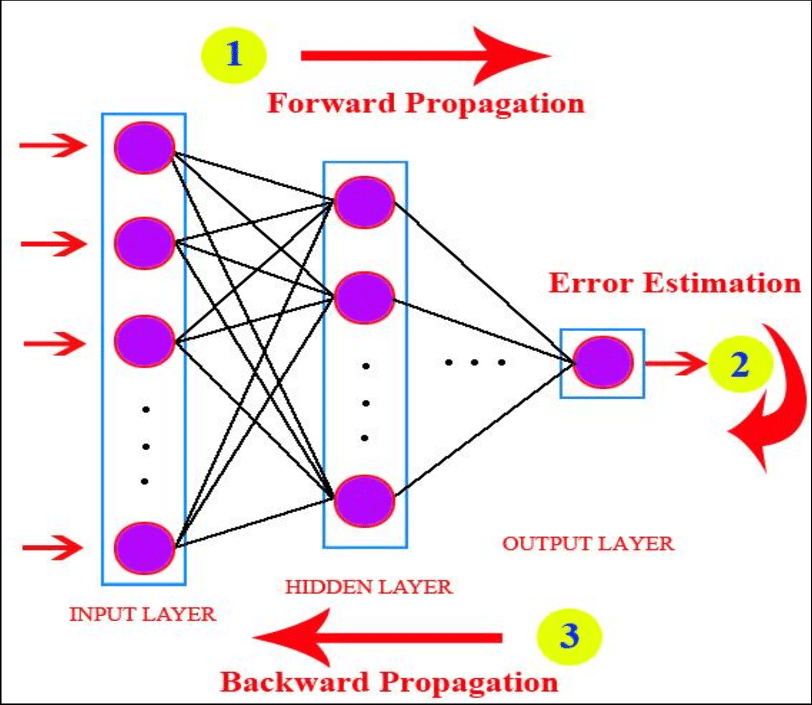
Simple GUI used to allow the user to communicate with the model. The picture uploaded to the model then ANN predicts which class the character belongs to. Finally, the prediction is printed on the screen.

A small demonstration is shown below:



Artificial neural networks (ANN)

An artificial Neural network is a mathematical model to mimics the brain of a human as it uses the concept of neurons to send information.

Every neural network has hidden layers to process the data from the input layer until it reaches the output layer. Every layer has some number of nodes like neurons and they are connected to each node in the next layer. The neural network can learn from the data by adjusting its weights. the weights are the lines that connect each node with the other nodes.   
there are two stages in the neural network, one is called forward propagation and it means the data are processed from the input layer to produce the output.   
the mechanism of the forward propagation is simple. The vector x is called the input vector and it has n dimensions. Each neuron is a function called an activation function. To transmit the input vector to the next layer, it is multiplied by the

A diagram of a diagram

Description automatically generatedweights of this layer. In the figure below, to forward propagate to the node h1 we need to multiply the output of i1 with w1 + the output of i2 times w3.

the input vector here is a 2D vector (0.1,0.5) and the node i1 takes that input using its activation function to produce the output that will be used in the hidden. by this mechanism, the values propagated until they reach the output layer. the output layer is the prediction of the neural network to some given values. for example, in our case, we use the neural network to detect the character from the image which is called optical character recognition. In the real world, the neural network starts with initially random weights and It tries to learn from the examples to adjust these weights. the algorithm used to do this is called the backward propagation. For each example you use forward propagation to see the prediction of the neural network then the true label to the example is fed to the network to know if its prediction is true or false. If the prediction is false, it adjusts its weights to be more accurate in the future. In backward propagation, an error formula is used. The simple one is A black text on a white background

Description automatically generatedthe true label minus the prediction squared

The goal is to minimize this function, which is affected by the weights of all the nodes as we calculate the output by the forward propagation. The idea of derivatives and chain rule comes here to update the weights. we see how much each weight changes the cost function by calculating the derivative of the error formula with respect to each weight. for example dE/dw1. Then we update the weight w1 by this formula

A black text on a white background

Description automatically generatedwe subtract the weight by how much it affects the error function, and that makes sense because we want to minimize the error function, and we can do this by decreasing the weights that corresponds to high error. By doing this for all the weights, the neural network can adjust its weights

Activation functions  
an activation function is used in ANN in each neuron, it helps the neural network to learn non-linearity and complex patterns. As discussed earlier in forward propagation, each neuron takes the input from the previous layers and then multiplies them by their weights, you can think of activations functions as function graphs that can be transformed into different shapes to fit the training examples. the transformations are done by multiplication and addition of weights. in this project, we use relu, linear, and SoftMax as activation functions. In each layer, the type of the activation function has to be specified.

RELU:

A blue line graph with numbers

Description automatically generatedrelu stands for Rectified Linear Unit. It returns the output if it is more than zero and zero elsewhere. It solves problems that arise in other activation functions such as saturation and sensitivity. The graph of the function is shown below

SoftMax:

A mathematical equation with numbers and symbols

Description automatically generatedSoftMax is used in multiclassification tasks as it outputs a predicted probability given the weights in the neural network. Usually, it exists in the final layer of the network. The predicted probability is not the actual probability of the classes, it is just an estimation of it. the equation of SoftMax is shown below:

For each class, the probability is given by exponentiation of the numerical prediction of the class in the last layer divided by the summation of the exponentiations of the other classes.

Linear activation function:

A graph with a green line

Description automatically generatedThe linear activation function is just the identity function f(x) =x. it maps the input to the output. usually, it is used as the last layer in regression neural networks.

Loss function:

The loss function or error function is the function that neural networks use to know if its prediction is true or false. Minimizing the loss function helps the neural network to learn from the training examples by adjusting its weights to make its predictions accurate in the future. The loss function used in multi-classification problems called categorical cross-entropy

categorical cross-entropy  
it is SoftMax activation plus cross-entropy. The equation of cross-entropy is shown below:

A diagram of a mathematical equation

Description automatically generatedA white lines on a black background

Description automatically generatedwhere it is the actual truth of the classification and si is the prediction probability of the neural network. In the categorical cross-entropy, the final layer passed to SoftMax then the cross-entropy.

As in multi-classification, one class is only true and its truth is one and the rest is zero. Therefore, its loss only remained making the equation   
A black and white image of a log

Description automatically generatedby taking derivatives to this function and minimizing it, the neural network can update the weights.

ADAM optimizer

A rainbow colored lines and dots

Description automatically generatedOptimization algorithms have been developed to make the minimization of the loss function faster. Searching for the weights that minimize the loss can take a lot of time and it is an essential task in the learning process of the neural network. One of these algorithms is adaptive moment estimation (ADAM). ADAM is a combination of gradient descent with momentum and the RMSP algorithm. The gradient descent is an algorithm used to find the direction we should take to minimize the function and how much we should move in that direction defined by the learning rate. The gradient descent problem is that it is moving in a zigzag way. Using momentum makes the steps more forward to the goal as shown in the graph below.

The idea of the momentum is to accelerate the gradient descent algorithm by taking into consideration the ‘exponentially weighted average’ of the gradients. Using averages makes the algorithm converge toward the minima at a faster pace. The hyperparameters of the ADAM are two decay rates to each algorithm (gradient descent with momentum and the RMSP algorithm) of the average of gradients and learning rate which is how much should we step to the direction of minimization.

Implementation Details

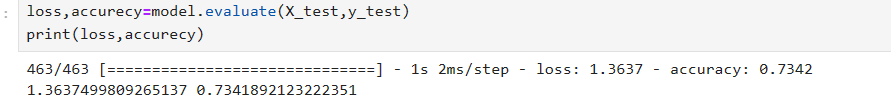
EMMIST characters dataset has been used to train a neural network. The dataset is divided into files (training and testing). The shape of the data set is 785 columns. The first column is the label of the training example. The rest of the columns are the pixels of the picture that contain the alphabet. The pictures are 28\*28 pixels and the pixels are represented in one row in the dataset 784 features. The pixels are in greyscale, each pixel has a value between 0 and 255. the questions of how many layers should be used, how many neurons in these layers, and which activation functions should be used, have no clear answers, and all done in the project is trying different compositions to reach the highest accuracy.

**The first trial(model 1):**

A screenshot of a graph

Description automatically generated a simple neural network has been examined with two layers only. The intuition behind using only two layers is that maybe the first layer learns the edges and the second layer learns the loops. Although this intuition is not accurate, it is a good way to start the first layer has 784 neurons the same as the number of pixels in the training examples. in that layer, the relu activation function has been used because it is the most used activation function in neural networks and it is known for its practical use. The second layer has 26 neurons because there was 26 class (each class has the lowercase and the uppercase of the same alphabet). The activation function used in this layer is SoftMax. As discussed, earlier SoftMax is used to estimate probabilities for classes and it is the best function for multi-classification problems. Categorical cross-entropy is used as the loss function because it is suited to multi-classification problems and has a high loss when the neural network misclassifies an example as less probable. Adam optimizer is used as an optimizer

to minimize the loss function. A small number of epochs was used (20).  
the graph of the accuracy and the loss is shown below

at first glance, the time the network took to learn was relatively big besides the small number of epochs. The graph of the training accuracy does not converge absolutely. Many factors should be the reason for that, one of them is the number of epochs is not big enough, or the learning rate is too high. lastly, maybe there is a lot of noise in the data. the accuracy of the model on the testing should be below:

the accuracy is 73% relatively good but another model should be examined to enhance it.

**The second trial(model1):**

A graph of training and loss

Description automatically generatedIn the second trial, normalization was used on the pixels before feeding it into the model. The idea of normalization is the process of scaling input features to a similar range. This can improve the performance and training speed of the neural network. The technique used to normalize the pixels is to divide each one by 255 as the range of each pixel is between zero and 255. By this, we make sure that all the pixels fall within a close range. Also, increasing the number of epochs to 50 to make sure that the training accuracy will converge. The training accuracy and training graph:

The graphs look smoother and the training accuracy is converged. For accuracy , it is better than the previous trial with an accuracy 0.89%

A close up of a text

Description automatically generated

**The third trial(model2):**

In the third trial, another technique was used for feature extraction to enhance the model. Histogram of oriented gradients (HOG) used for this task. HOG is used for highlighting edges and boundaries in the picture and that can help the model to detect the classes(alphabets) more accurately.  
the ideal of HOG is to calculate the gradient of each pixel in the cell by applying a kernel on the y-axis and another one on the x-axis. Also, the direction of the gradient is calculated. The image is divided into small cells, typically of size 8x8 pixels. The HOG descriptor is calculated independently for each cell. For each cell, a histogram of gradient orientations is created. The orientations are quantized into bins, and each gradient contributes to the corresponding bin in the histogram based on its orientation. Then a normalization is applied for the bins in each cell.   
an example is shown below:

A close-up of images

Description automatically generated

A close-up of a logo

Description automatically generatedTo determine how many features will result from the 28\*28 picture. The picture is divided into 8\*8 cells. each cell will have 9 features independent from the other cells. To calculate the number of 8\*8 cells, the equation:

So, the number of cells will be (28/8) \*(28/8) =9. So, the number of features will be the number of cells times the length of features in each cell and that will be equal to 9\*9 = 81 features.

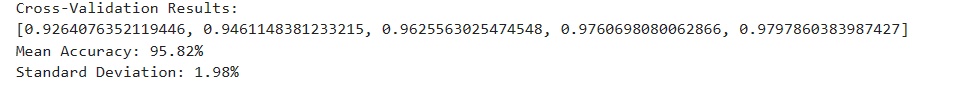
By this, we decline the number of neurons in the input layer to 81 instead of 784. One hidden layer is added to the previous trial with 128 neurons and relu activation function. The intuition behind using the hog and adding one more layer is that HOG will help the neural network focus on the edges and the loops without the other noise in the character picture. Adding another layer also may help the model to learn more different shapes and loops and identify the patterns in the training examples. Therefore, the final architecture is two layers with relu activation function and one final layer with SoftMax to output the probabilities of the classification.

**Metrics for comparisons between the models:**

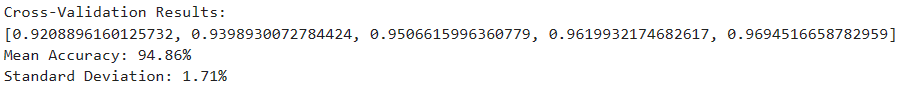
**Cross-validation:**

The idea of cross-validation is to split the training dataset into splits. Then use a ratio of these splits in training the model and the remaining splits in testing. Doing that for some iterations then taking the average of the results. This technique is more accurate than the normal accuracy which is to split the data in only two splits one for training and one for testing then taking the percentage of the number of accurate predictions on the testing set. the problem with the traditional technique is that the results are not generalized, but in cross-validation taking different sets for training and testing gives more generalized results about how well the model is doing in predictions. in the code, 5 splits have been used, four are chosen for training and one for testing in five iterations.

**For model 1 (trial2 )**



**For model 2**

****

Model 1 has better accuracy on average indicating that it is better in learning from the data but model 2 is more consistent in performance across different test sets.

**Weighted precision:**

Precision is how many predictions are positive when the prediction is positive. For example, if the model prediction that the picture is “A” how many times when the model predicts “A” its true label is “A”

The formula for precision for each class i

A math equations and numbers

Description automatically generated with medium confidenceIn multi-classification, a different version of precision is used called weighted precision. For each class we calculate its precision then the average is calculated for all the classes. The equation is

The idea behind weighted precision is to give more importance to classes with larger support (more instances) in the dataset. This helps address the impact of class imbalance on the evaluation metric.

For model1, the precision was 0.90 while model2 was 0.88. which means that for

Model 1

Out of all instances predicted as positive by Model 1, 90% are true positives (correctly predicted positives), and 10% are false positives (instances predicted as positive but are negative).

Model 2

Out of all instances predicted as positive by Model 2, 88% are true positives, and 12% are false positives.

**Weighted Recall:**

Recall, also known as sensitivity or true positive rate, is a metric used to evaluate the ability of a classification model to correctly identify positive instances.

The equation of recall is

A close up of a text

Description automatically generated

The idea of weighted recall is the same as weighted precision.

The equation is as follows

A math equations on a white background

Description automatically generatedboth models have the same recall which is 88%. the recall value of 88% indicates that 88% of positive instances are correctly captured by the models, and 12% of positive instances are falsely classified as negative (false negatives).

**F1 score:**

The F1 score is a metric used in classification tasks that combines precision and recall into a single value. It provides a balanced measure of a model's performance by considering both false positives and false negatives. The F1 score is particularly useful when the class distribution is imbalanced.

The equation is as follows

A close up of a sign

Description automatically generated

The F1 score for Model 1 is 0.89 and for Model 2 is 0.88. The F1 score of 0.88 suggests that Model 2 also has a good balance between precision and recall, but it may be slightly less effective than Model 1.

**Conclusion**:

Model 1 outperforms Model 2 across precision, recall, and F1 score. Model 1 is more balanced in terms of precision and recall, resulting in a higher F1 score. If precision is a crucial factor, Model 1 may be preferred; otherwise, both models exhibit reasonably good performance. Although the two models have good accuracy, it turns out that adding HOG was not very beneficial for enhancing the performance of the model. One of the reasons that some details were not captured by local gradient by HOG

Recommendations and future work

The models in this project were trained on pictures of a single character and cannot be used in classifying a whole sentence or mixture of characters. Also, this model trained only on the English alphabet and there is no support for any other language. Also, investigate methods to enhance the model's performance on noisy or degraded images. This could involve exploring denoising techniques or developing strategies to handle low-quality scans or images with artifacts. A convolution neural network (CNN) can have a better performance than an ordinary artificial neural network.