

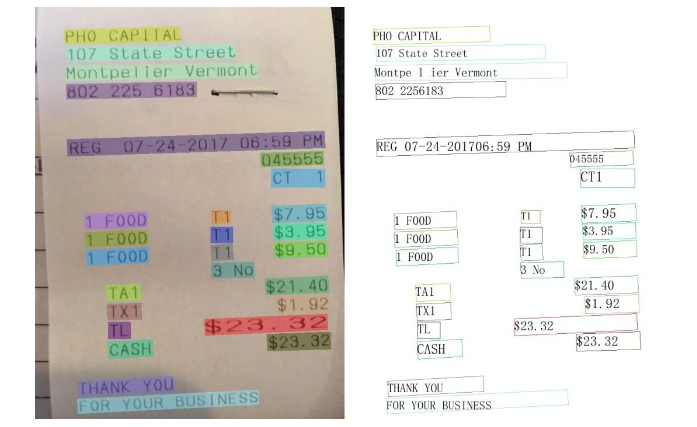
Artificial Intelligence Documentation

|  |
| --- |
|  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Written By:   |  |  |  | | --- | --- | --- | | **NAME** | **Section** | **ID** | | **Emad Mostafa Srag** | CS | 202000580 | | **Hana Mohamed Zein** | CS | 202001039 | | **Yasmin Hamdy** | CS | 201900958 | | **Moatasem Mohamed** | CS | 202000928 | | **Aya Raafat Metwalli** | CS | 202000184 | | **Mohamed Ali Mosleh** | CS | 202000804 | |

Optical Character Recognition "OCR"

What is OCR??

OCR (optical character recognition) is the use of technology to distinguish printed or handwritten text characters inside digital images of physical documents, such as a scanned paper document. The basic process of OCR involves examining the text of a document and translating the characters into code that can be used for data processing. OCR is sometimes also referred to as text recognition. Hardware, such as an optical scanner or specialized circuit board, is used to copy or read text while software typically handles the advanced processing. Software can also take advantage of artificial intelligence (AI) to implement more advanced methods of intelligent character recognition (ICR), like identifying languages or styles of handwriting. 

What is Scene Text Recognition (STR)?

In computer vision, machines can read text in natural scenes by first detecting text regions, cropping those regions, and subsequently recognizing text in those regions. The vision task of recognizing text from the cropped regions is called Scene Text Recognition (STR). STR makes it possible to read road signs, billboards, logos, and printed objects such as text on shirts, paper bills, etc. STR applications include practical use cases such as self-driving cars, augmented reality, retail analysis, education, devices for the visually impaired, and others.



How Artificial Intelligence Gives OCR a Boost?

Artificial intelligence is transforming the capabilities of optical character recognition (OCR) tools. An area of computer vision, OCR processes images of text and converts that text into machine-readable forms. In other words, it takes handwritten or typed text within physical documents and converts them into digital formats.

Process and function of Optical Character Recognition (OCR)

Handwrite recognition is one of the types of Optical Character Recognition (OCR). OCR is identification of text, which may be printed or hand-written. In OCR, the document is captured via camera as image and can be converted to desired formats like PDFs. Then the file is fed to the algorithm for character recognition

Optical Character Recognition (OCR)

Printed Character Recognition

Handwrite recognition

Feature extraction

Classification

Offline Character Recognition

Online Character Recognition

Image source

Pre-processing

segmentation

Classification

Feature extraction

1.Image source

This phase comes in offline hand-written character recognition. Image source can be from any digitized tool. A scanner or a camera captures the image and is sent to next phase.

2. Pre-processing:

that improves the quality of image and hence increases the accuracy of image. For the hand-written character recognition process, the following pre-processing techniques are followed.

1. Noise-removal:

It is the process of removing noise from image. This also refers to smoothening the image by reducing the unwanted signals in the image. There exist many algorithms to remove noises in the image. Some of them are Gaussian filtering method, Min-max filtering method, Median filter etc.

1. Skew correction:

correcting the skew (deviation of the baseline from the horizontal direction and the slant (deviation of average near-vertical strokes from the vertical direction is an important preprocessing step. Typically, first the skew angle and then the slant angle are corrected [1], [3], [7]. In some cases, however, it might be more convenient to first correct the slant and then the skew [8].

c) Binarization:

It is a mechanism of converting grayscale or colored image to binary image. Binary images contain only 0’s and 1’s. The pixels in images are partitioned to 0’s or 1’s based on some constant value. If the pixel value is less than the constant, it is replaced with 0 or else with 1.

d) Morphological operation:

This is the process of increasing or decreasing the size of an image. This is done mainly because the algorithm would expect the constant image size. To increase the size of an image, we can add pixels to the boundary of image. To decrease the size of an image, we can remove pixels from the boundary of image.

3. Segmentation:

Segmentation is a mechanism that extracts individual characters in the image. There exist two types of segmentation. They are Implicit segmentation and Explicit segmentation. In implicit segmentation the words are recognized without segmentation process. But in explicit segmentation, words are predicted by extracting individual character.

4. Feature-extraction

This is the important phase in the recognition process, and the algorithm of recognition starts from here. Each character contains its own features. It contains group of rules where each rule explains feature of a character. Extraction of such features is done in this phase. 5. Classification By this time, the training would have completed, and the testing of input data starts. The testing data would pass all the above process and the varying probabilities are assigned to the matching rules. The rule with highest probability is selected and the corresponding class-label is made recognized character.

5. Classification

By this time, the training would have completed, and the testing of input data starts. The testing data would pass all the above process and the varying probabilities are assigned to the matching rules. The rule with highest probability is selected and the corresponding class-label is made recognized character

Applications of Optical Character Recognition in real life:

**Google Translate:**

* One of the most widely used OCR applications stems from[Google Translate’s OCR addition](https://www.cnet.com/tech/mobile/google-translate-for-android-adds-ocr/), which revolutionized the way we communicate in different languages. Instead of typing text into the app like before, you can simply point your mobile phone camera at text, and it will be scanned into Google Translate via OCR.
* This is extremely useful, and it can sometimes be the only viable method of translation. If you’re in a foreign country and you see a signboard, even with a normal translation app, you won’t necessarily know what it means if you can’t speak the language. Some languages don’t use the same script style as English, so you likely won’t even be able to input the correct characters into the translation app. Bringing OCR to the translation game changes everything — you don’t need to know the language to understand it. Google’s advanced OCR can identify and parse the language into its system, providing instant identification of what has been written. For millions of travelers worldwide, this has been a game-changer, especially in countries with vastly differing language scripts (China, Japan, Russia, India, etc.).

**Cambridge Assessment:**

* is another company successfully using OCR to automate complex tasks. Cambridge Assessments runs some of the world’s biggest examinations, including GCSEs, IGCSEs, and Cambridge A Levels. They process over one million physical papers per year. Examination papers are handwritten by students, and Cambridge examiners work remotely from all over the world, making it infeasible to send physical copies of answer scripts. To solve their problem, Cambridge Assessments turned to technology.
* Simple scanning and replicating won’t really do the job: although it transmits the papers electronically to the examiners, the process makes marking much harder. Examiners need to manually go through each answer script and have to sift out the relevant information. Instead, [Cambridge Assessments uses OCR technology](https://www.cambridgeinternational.org/exam-administration/results/marking-and-grading/) to scan each paper that passes through their system. This allows for proper documentation and sorting of answers. For example, with OCR technology, papers can be scanned into the system, and computers can group together sets of papers based on age range, location, examination type, examination level, and more. Humans don’t need to physically sit at sorting centers to group papers together—OCR makes it possible via computers. Additionally, OCR allows answers to be separated by question. For example, if an examiner wants to mark “question 2” from a given paper, OCR technology can scan and separate out all of the “question 2s” from all the papers so that the examiner can mark them all at once. This is far more efficient and faster than having an examiner manually open each document, scroll to question 2, and then mark all of them one by one.

**Stadtwerke München:**

* a communal company owned by the city of Munich, was one of the first major government-led organizations to adopt a novel form of OCR—[license plate scanning](https://anyline.com/news/city-parking-stadtwerke-muenchen/). The company offers public services such as public transport and parking management, and as such, parking inspection became an important part of their work. However, manual parking inspection can be quite a chore—to ensure accuracy, parking inspectors were made to input the license plate number into the system twice, manually. Not only was this cumbersome, it was also a waste of resources that could have easily been deployed elsewhere.
* The city of Munich found their solution in Any line, an OCR app that allowed parking inspectors to accurately input license plate numbers into their system in under one second. This allows them to save a lot of time and effort during parking management. A spokesperson for Stadtwerke München said OCR has “made it much faster for our parking attendants to collect license plate data. This efficiency improvement at the base of our parking control operations will help us to improve parking control within Munich.”

**The front of a car

Description automatically generated with medium confidence**

**OCR Applications in Banking:**

* The banking industry is deemed one of the largest consumers of OCR recognition apps as it helps enhance security, improve data management, optimize risk management, and enhance customer experience. Before applying OCR technology, most banking documents were physical, including customer records, checks, bank statements, and others. Using an OCR recognition solution, it becomes possible to digitize and store even older documents in databases. OCR technology has also completely revolutionized the banking industry by:
* Providing easy verification: OCR allows a real-time verification of money deposit checks and a signature by scanning them using an OCR-based application. An example of this can be seen in mobile banking apps, where checks can be deposited digitally and processed within days through OCR-based check depositing features.
* Enhancing security: The electronic deposition of checks through OCR technology results in fraud prevention and increasingly secure transactions, fostering a better user experience. OCR can use the character reader count and machine learning methods to detect forged documents.

**Graphical user interface

Description automatically generated**

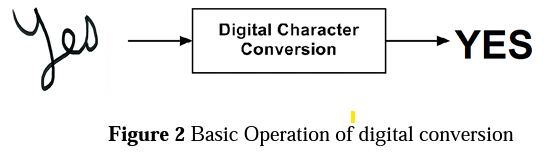
* **OCR Use Cases in Healthcare OCR machine:**
* learning has proved to be beneficial for the healthcare industry. In the healthcare sector, OCR technology allows patient medical histories to be accessed digitally by patients and doctors alike. In addition, patient records, including their X-rays, treatments, tests, hospital records, and insurance payments, can easily be scanned, searched, and stored using OCR full form methods to digitize records and read labels with cameras. Thus, optical character recognition helps streamline the workflow and reduce manual work at hospitals while keeping the records up to date.

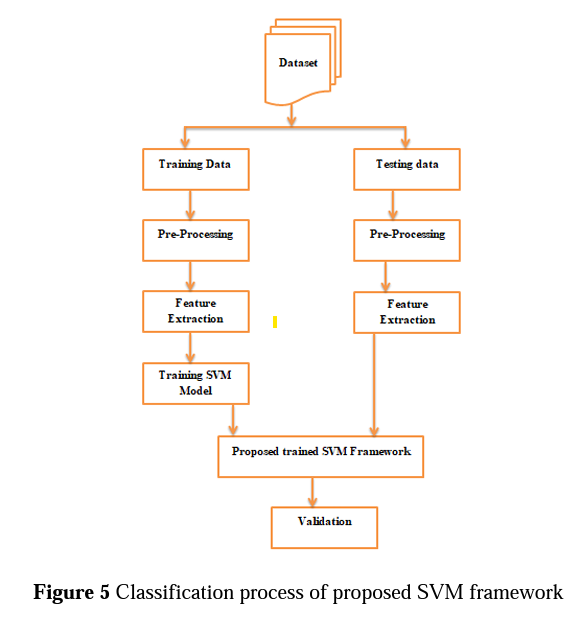
# Graphical user interface, text, application Description automatically generated

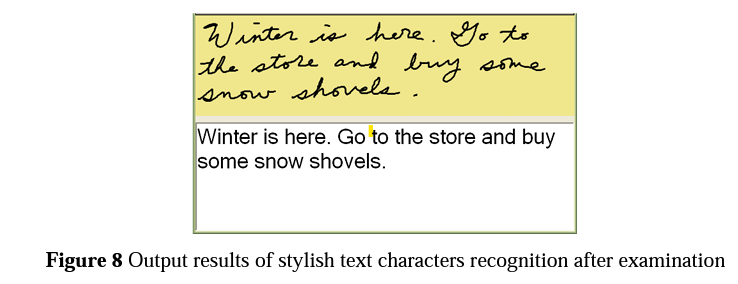
Academic publications (papers):

* First paper

Image:







Description:

the research paper is comparing statistical approach support vector machine (SVM) classifiers network method with statistical, template matching, structural pattern recognition, and graphical methods. It has proved Statistical SVM for OCR system performance that is providing a good result that is configured with machine learning approach.

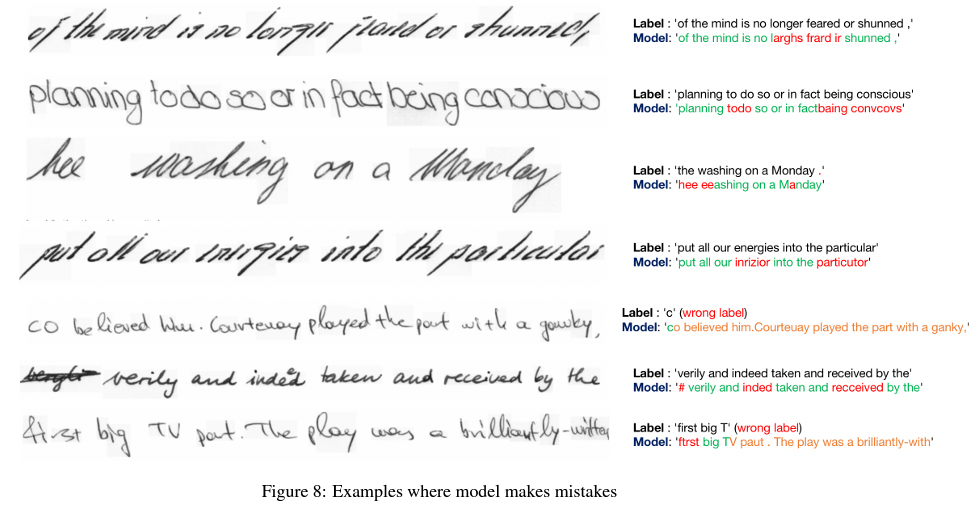
Link:

https://irojournals.com/itdw/V3/I2/03.pdf

* Second paper

Image:





Description:

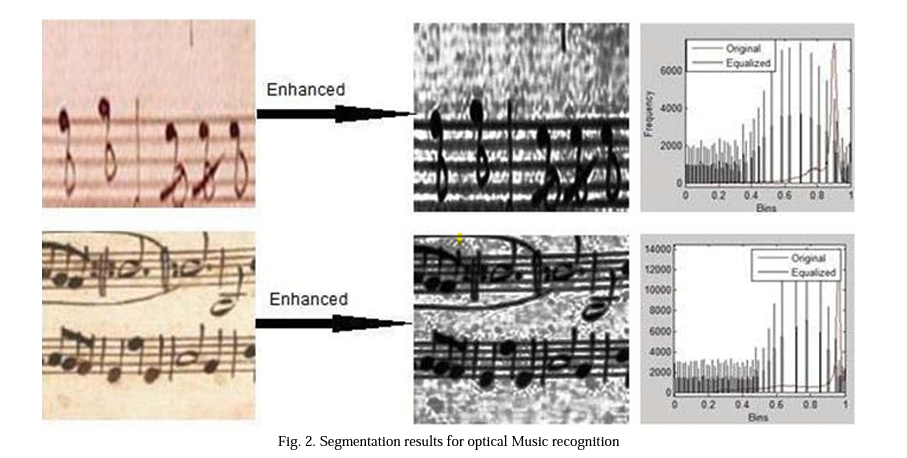
In this paper, we propose a CNN based architecture that bridges this gap. Thier work, Easter2.0, is composed of multiple layers of 1D Convolution, Batch Normalization, ReLU, Dropout, Dense Residual connection, Squeeze-and-Excitation module and make use of Connectionist Temporal Classification (CTC) loss. In addition to the Easter2.0 architecture, we propose a simple and effective data augmentation technique ’Tiling and Corruption (TACo)’ relevant for the task of HTR/OCR. Our work achieves state-of-the-art results on IAM handwriting database when trained using only publicly available training data.

Link:

<https://paperswithcode.com/paper/easter2-0-improving-convolutional-models-for>

* Third paper:

Image:



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>

* Fourth paper:

Image:



Description:

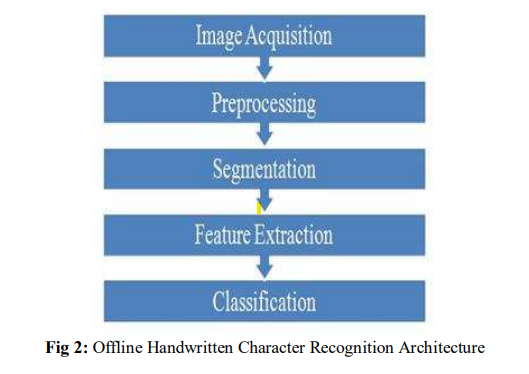
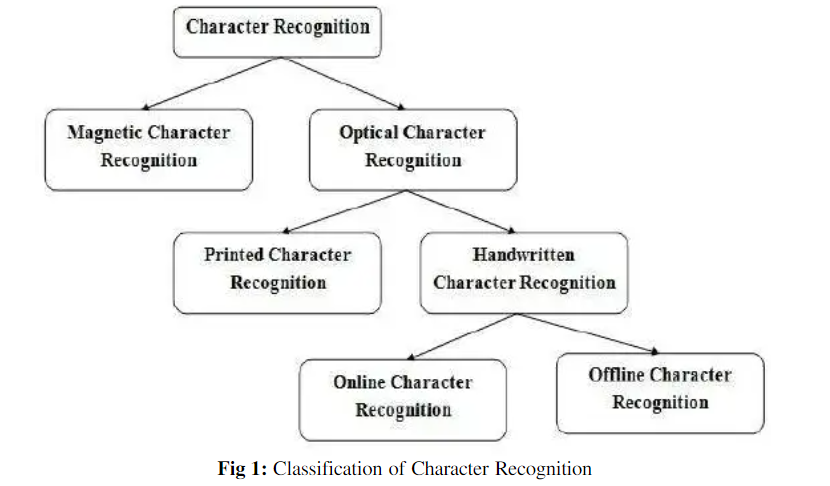
Optical character recognition refers to the branch of computer science that involves reading text from paper and translating the images into a form that the computer can manipulate (for example, into ASCII codes). An OCR system enables you to take a book or a magazine article, feed it directly into an electronic computer file, and then edit the file using a word processor.All OCR systems include an optical scanner for reading text, and sophisticated software for analyzing images. Most OCR systems use a combination of hardware (specialized circuit boards) and software to recognize characters, although some inexpensive systems do it entirely through software. Advanced OCR systems can read text in large variety of fonts, but they still have difficulty with handwritten text.It is the mechanical or electronic translation of scanned images of handwritten, typewritten or printed text into machine-encoded text. It is widely used to convert books and documents into electronic files, to computerize a record-keeping system in an office, or to publish the text on a website.

Link:

<https://www.academia.edu/download/47097399/D017222226.pdf>

* Fifth paper:

Image:

Description:

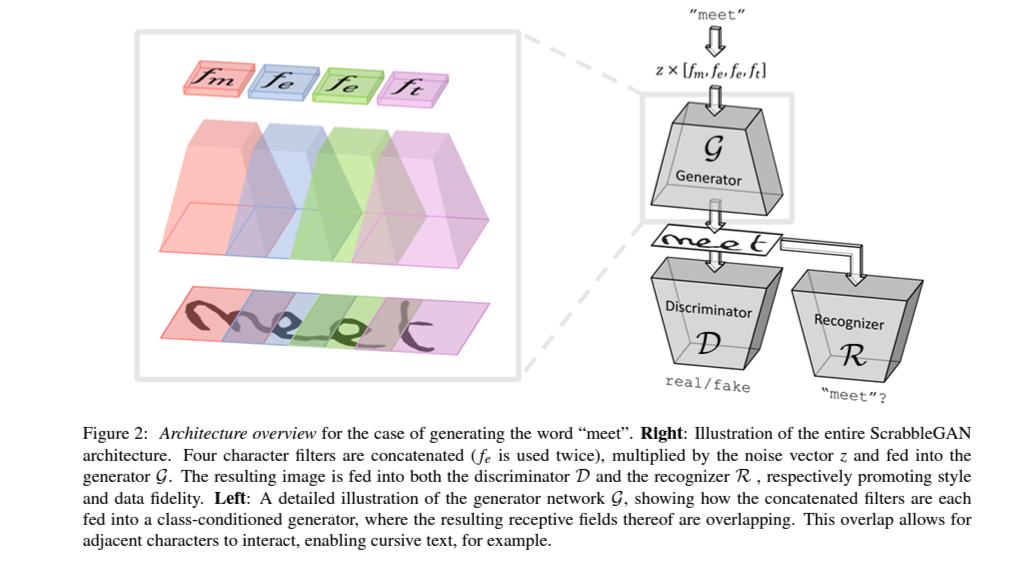
This paper presents a detailed review of Offline Handwritten Character Recognition. HCR is an optical character recognition, which convert the human readable character to machine readable format. In HCR, to attain 99% accuracy is very difficult.

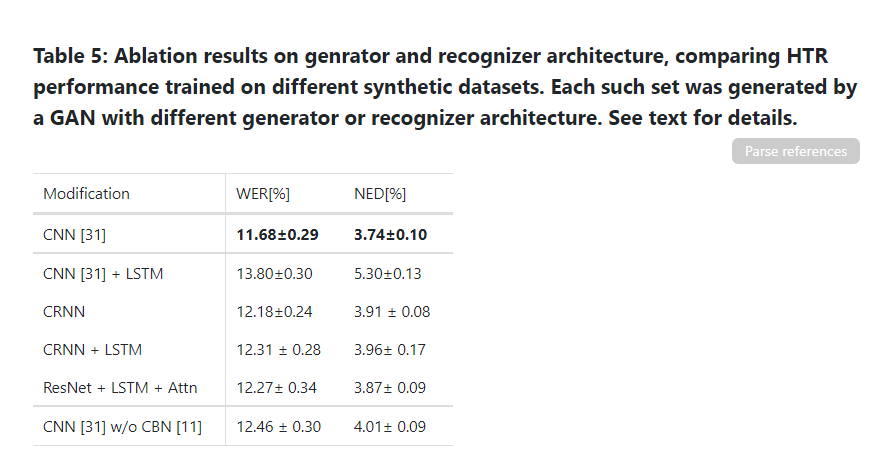
Link:

<https://www.academia.edu/26823845/Handwritten_Character_Recognition_A_Comprehensive_Review_on_Geometrical_Analysis?from=cover_page>

* Sixth paper:

Image:





Description:

systems performance have improved significantly in the deep learning era. This is especially true for handwritten text recognition (HTR), where each author has a unique style, unlike printed text, where the variation is smaller by design. That said, deep learning based HTR is limited, as in every other task, by the number of training examples. Gathering data is a challenging and costly task, and even more so, the labeling task that follows, of which we focus here. One possible approach to reduce the burden of data annotation is semisupervised learning. Semi supervised methods use, in addition to labeled data, some unlabeled samples to improve performance, compared to fully supervised ones. Consequently, such methods may adapt to unseen images during test time.

Link:

https://paperswithcode.com/paper/scrabblegan-semi-supervised-varying-length

Dataset

Train:

https://www.kaggle.com/datasets/crawford/emnist?select=emnist-letters-train.csv

Test:

https://www.kaggle.com/datasets/crawford/emnist?select=emnist-letters-test.csv

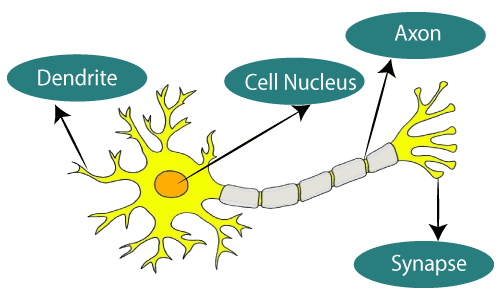
Mapping:

<https://www.kaggle.com/datasets/crawford/emnist?select=emnist-letters-mapping.txt>

# what are neural networks.?!

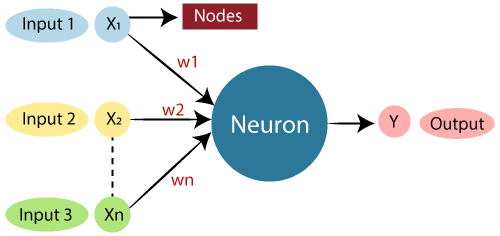
Neural networks reflect the behavior of the human brain, allowing computer programs to recognize patterns and solve common problems in the fields of AI, machine learning, and deep learning.

The term ****Artificial Neural Network**** is derived from Biological neural networks that develop the structure of a human brain. Similar to the human brain that has neurons interconnected to one another, artificial neural networks also have neurons that are interconnected to one another in various layers of the networks. These neurons are known as nodes.

Biological Neural Network

|  |  |
| --- | --- |
| Biological Neural Network | Artificial Neural Network |
| Dendrites | Inputs |
| Cell nucleus | Nodes |
| Synapse | Weights |
| Axon | Output |

The typical Artificial Neural Network looks something like the given figure.



## **The architecture of an artificial neural network:**

Lets us look at various types of layers available in an artificial neural network.

### **Input Layer:**

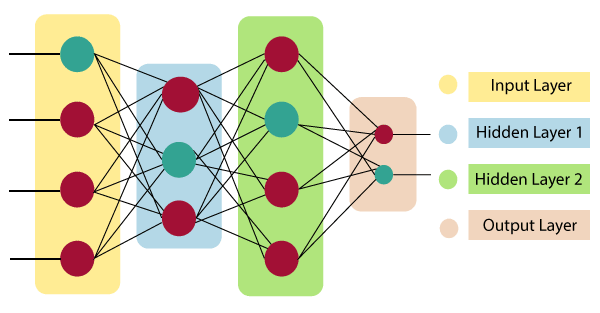
As the name suggests, it accepts inputs in several different formats provided by the programmer.

### **Hidden Layer:**

The hidden layer presents in-between input and output layers. It performs all the calculations to find hidden features and patterns.

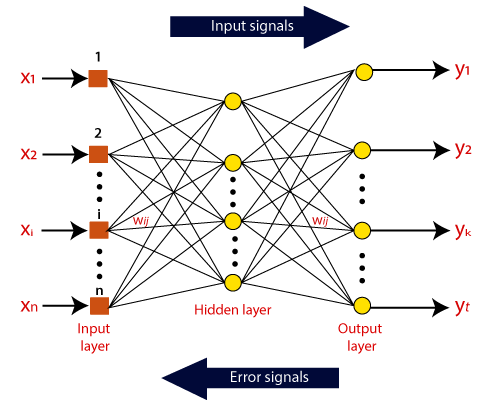
### **Output Layer:**

The input goes through a series of transformations using the hidden layer, which finally results in output that is conveyed using this layer.



## **How do artificial neural networks work?**

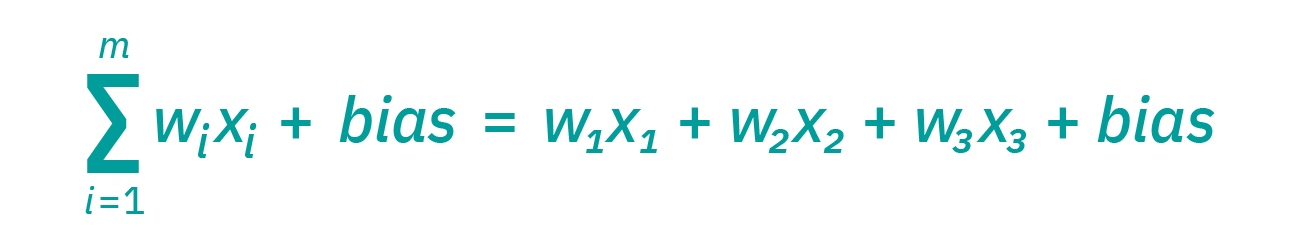
Artificial Neural Network can be best represented as a weighted directed graph, where the artificial neurons form the nodes. The association between the neurons outputs and neuron inputs can be viewed as the directed edges with weights. The Artificial Neural Network receives the input signal from the external source in the form of a pattern and image in the form of a vector. These inputs are then mathematically assigned by the notations x(n) for every n number of inputs.



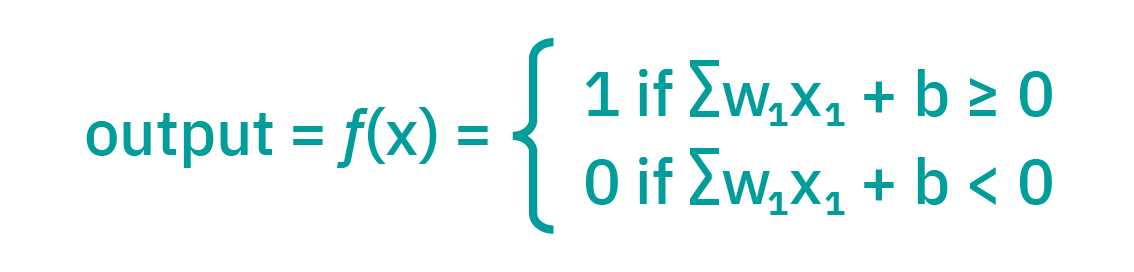
Afterward, each of the input is multiplied by its corresponding weights ( these weights are the details utilized by the artificial neural networks to solve a specific problem ). In general terms, these weights normally represent the strength of the interconnection between neurons inside the artificial neural network. All the weighted inputs are summarized inside the computing unit.

If the weighted sum is equal to zero, then bias is added to make the output non-zero or something else to scale up to the system's response. Bias has the same input, and weight equals to 1. Here the total of weighted inputs can be in the range of 0 to positive infinity. Here, to keep the response in the limits of the desired value, a certain maximum value is benchmarked, and the total of weighted inputs is passed through the activation function.

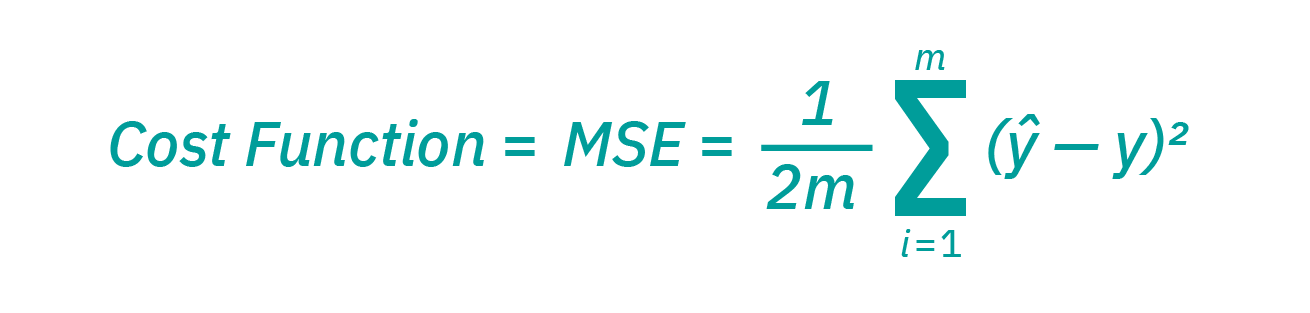
### **The formula would look something like this:**



### **output:**



### **cost function:**



## **Convolutional neural network(CNN):**

In deep learning, a convolutional neural network (CNN, or ConvNet) is a class of artificial neural network (ANN), most commonly applied to analyze visual imagery.

CNNs are also known as Shift Invariant or Space Invariant Artificial Neural Networks (SIANN), based on the shared-weight architecture of the convolution kernels or filters that slide along input features and provide translation-equivariant responses known as feature maps.

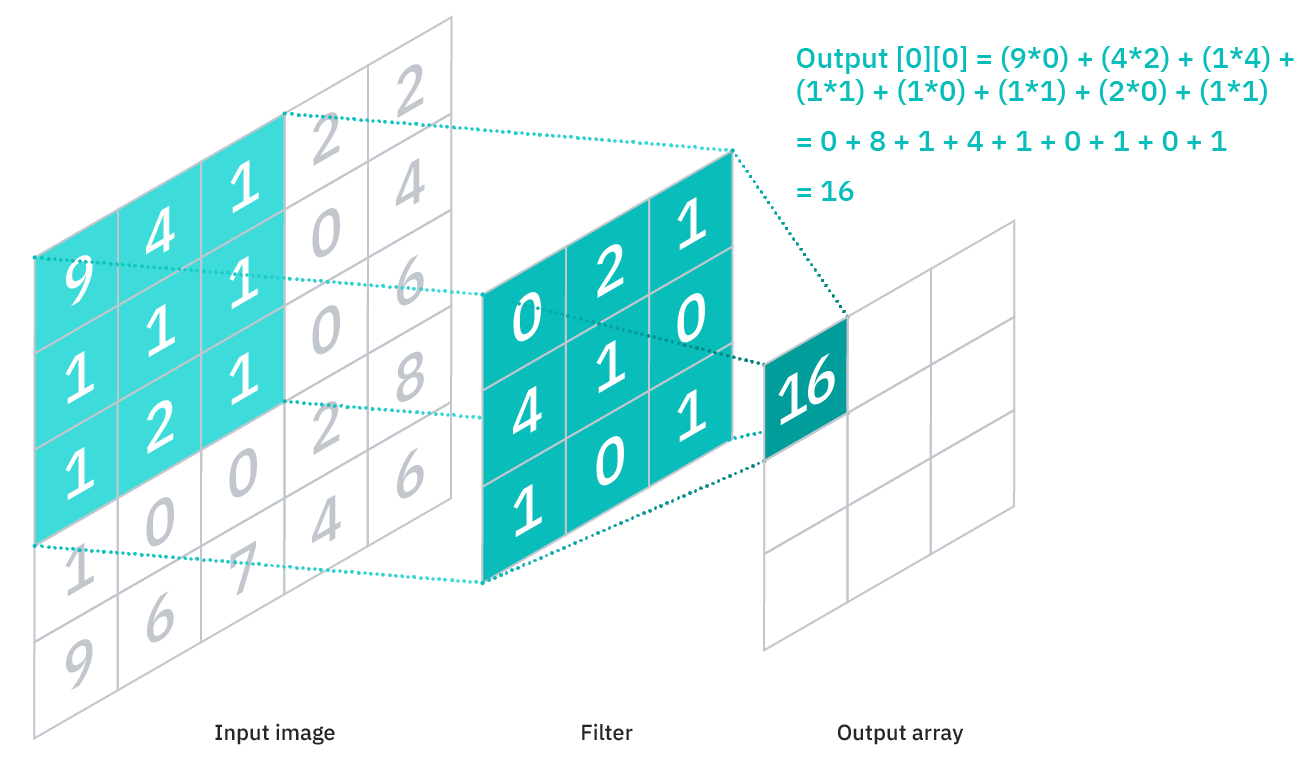
Counter-intuitively, most convolutional neural networks are not invariant to translation, due to the down sampling operation they apply to the input.

They have applications in…

* Convolutional layer
* Pooling layer
* flatten layer

### ***Convolutional Layer*:**

The convolutional layer is the core building block of a CNN, and it is where the majority of computation occurs. It requires a few components, which are input data, a filter, and a feature map.



### ***Pooling Layer*:**

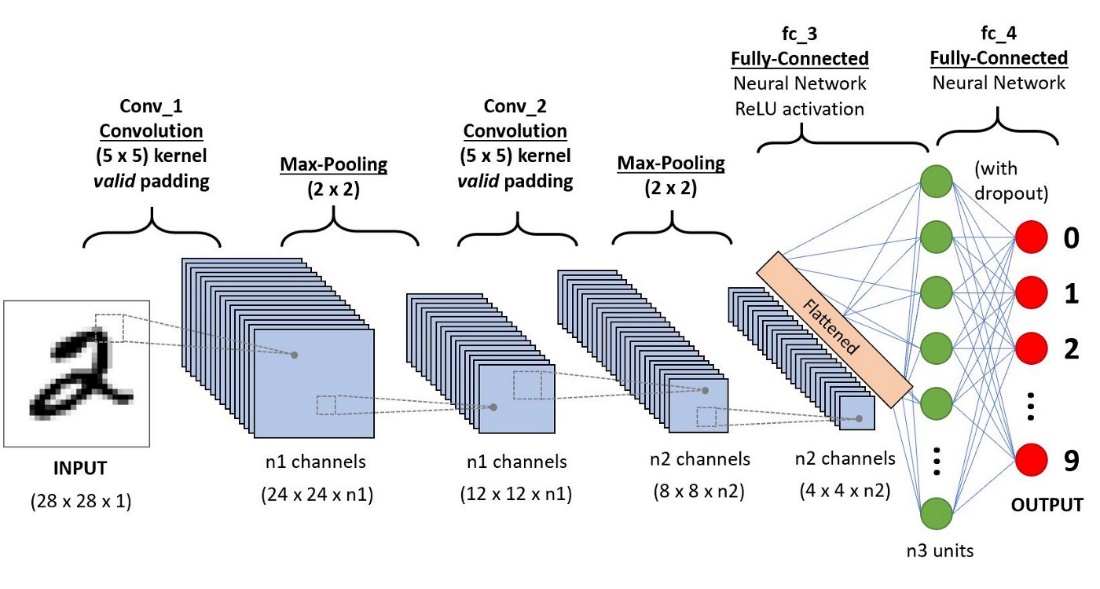
Pooling layers, also known as down sampling, conducts dimensionality reduction, reducing the number of parameters in the input. Similar to the convolutional layer, the pooling operation sweeps a filter across the entire input, but the difference is that this filter does not have any weights. Instead, the kernel applies an aggregation function to the values within the receptive field, populating the output array. There are two main types of pooling:

* Max pooling: As the filter moves across the input, it selects the pixel with the maximum value to send to the output array. As an aside, this approach tends to be used more often compared to average pooling.
* Average pooling: As the filter moves across the input, it calculates the average value within the receptive field to send to the output array.

### ***Fully-Connected Layer*:**

The name of the full-connected layer aptly describes itself. As mentioned earlier, the pixel values of the input image are not directly connected to the output layer in partially connected layers. However, in the fully-connected layer, each node in the output layer connects directly to a node in the previous layer.

This layer performs the task of classification based on the features extracted through the previous layers and their different filters. While convolutional and pooling layers tend to use ReLu functions, FC layers usually leverage a soft max activation function to classify inputs appropriately, producing a probability from 0 to 1.



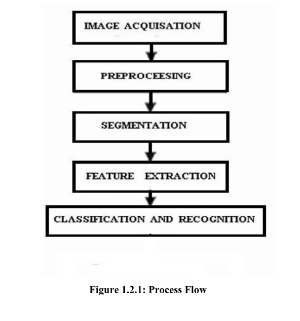
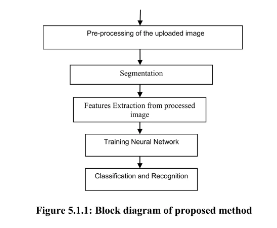
## **ANN vs CNN:**

Graphical user interface, text, application, email

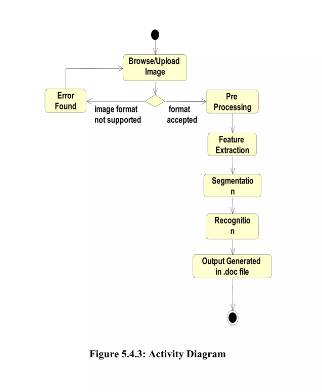
Description automatically generated

DIAGRAMS

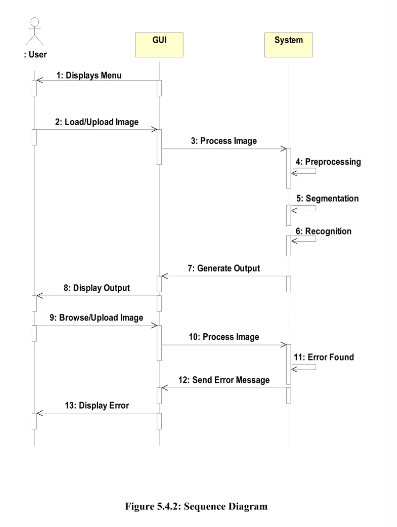
1. Block Diagram



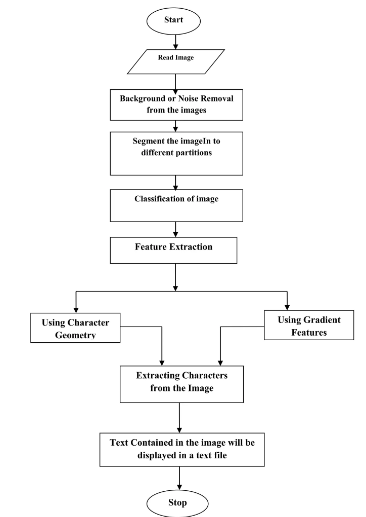
1. Activity Diagram



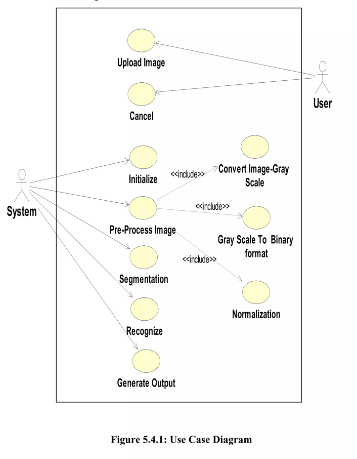
1. Sequence diagram



1. Flowchart



1. Use case



model

Handwritten alphabet recognition model:

ANN and CNN

There are specific instances in which ANN could be preferred over CNN and vice versa.

They are both unique in how they work mathematically, and this causes them to be better

at solving specific problems.

In general, CNN tends to be a more powerful and accurate way of solving classification

problems.

ANN is still dominant for problems where datasets are limited, and image inputs are not necessary.

Let’s go through the steps of creating, training, and testing the model.

We used all the letters to train our model with 88800 images, we split the images of each letter

into training and testing. We use almost 83% of the images to train and 17% to test.

The First Step Preparing the data :

We define A variable train and test to load the dataset then we split the data into features and target using .Iloc so we can put the features in X and the target in y.

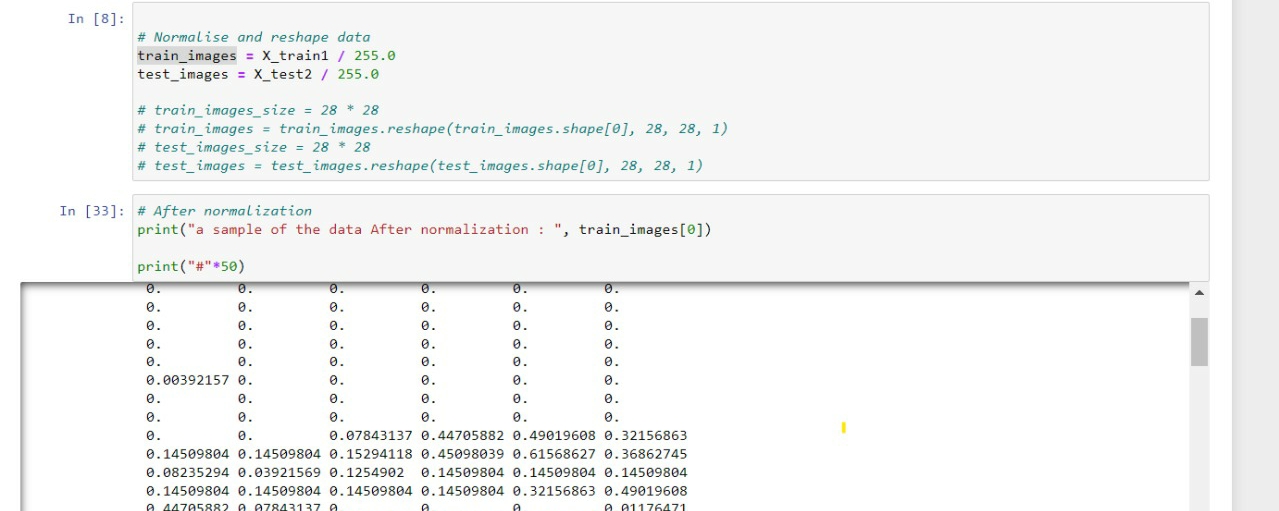


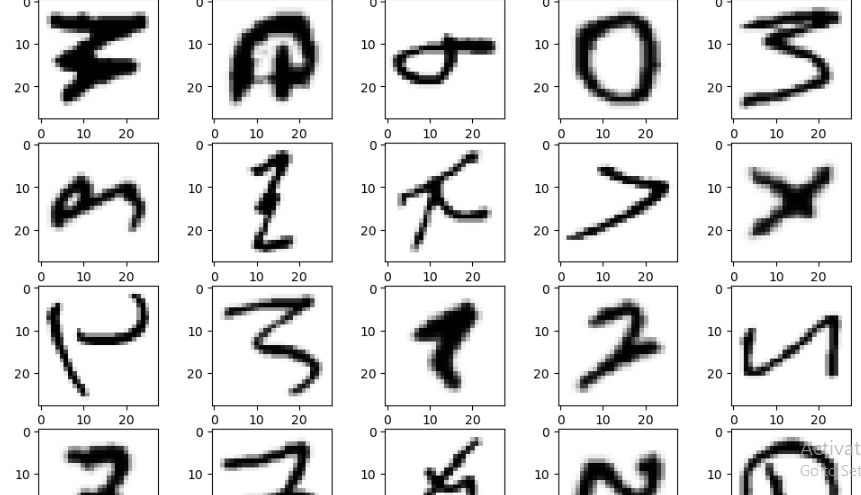
After that for training and testing our model and see how it performs we needed to split the data into training and testing with train\_test\_split().



Then we wanted to see some samples of our dataset that’s why we used matplotlib to do some visualization.

Then we divided the X\_train and X\_test by 255 to normalize them and by this the computation becomes easier and faster since all the numbers became in range of 0 and 1.





Then we reshape the data before entering the model because NN train faster on small images A larger input image requires the neural network to learn from four times as many pixels, and this increase the training time for the architecture.

Second Step Building the model:

Sequential() is to create a sequential model because we have many layers that

neurons of each layer are connected to the neurons of the next layer.

After that we created 5 Dense layers.

1st Layer:

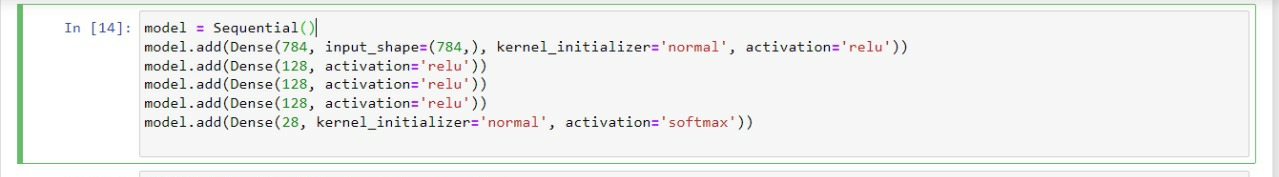
For this, we use input shape image of (784,) with 784, kernel initializer ‘normal’ Activation function in this layer is ReLu that will output the input directly if it is positive, otherwise, it will output zero. So if the pixel is contribute in an important feature that will remain, otherwise it will be discard and the output=0.

2nd Layer, 3rd Layer, 4th Layer:

As shown Below.

5th Layer:

For this, we use output shape of (28) because this is the number of the letters in our classifications, Activation function in this layer is softmax. The softmax function is used as the activation function that predict a multinomial probability distribution. The output of a Softmax is a vector (say v ) with probabilities of each possible outcome.



Third Step: Training a network

It is a process of finding weights in fully connected layers which minimize differences between output predictions and given ground truth labels on a training dataset.

A model performance under weights is calculated by a loss function through forward propagation on a training dataset, and learnable parameters weights, are updated according to the loss value through an optimization algorithm called backpropagation and gradient descent, among others loss function, measures the compatibility between output predictions of the network through forward propagation and given ground truth labels.

We compile the model with 'categorical\_crossentropy’ loss function is used for multiclass classification model where there are two or more output labels.

The optimizer is an algorithm that is used to change the attributes of the neural

network such as weights and bias. Adam optimization is a stochastic gradient descent

method that is based on adaptive estimation of first-order and second-order moments.

The optimizer generates the hyperparameters and reached the best values for the best

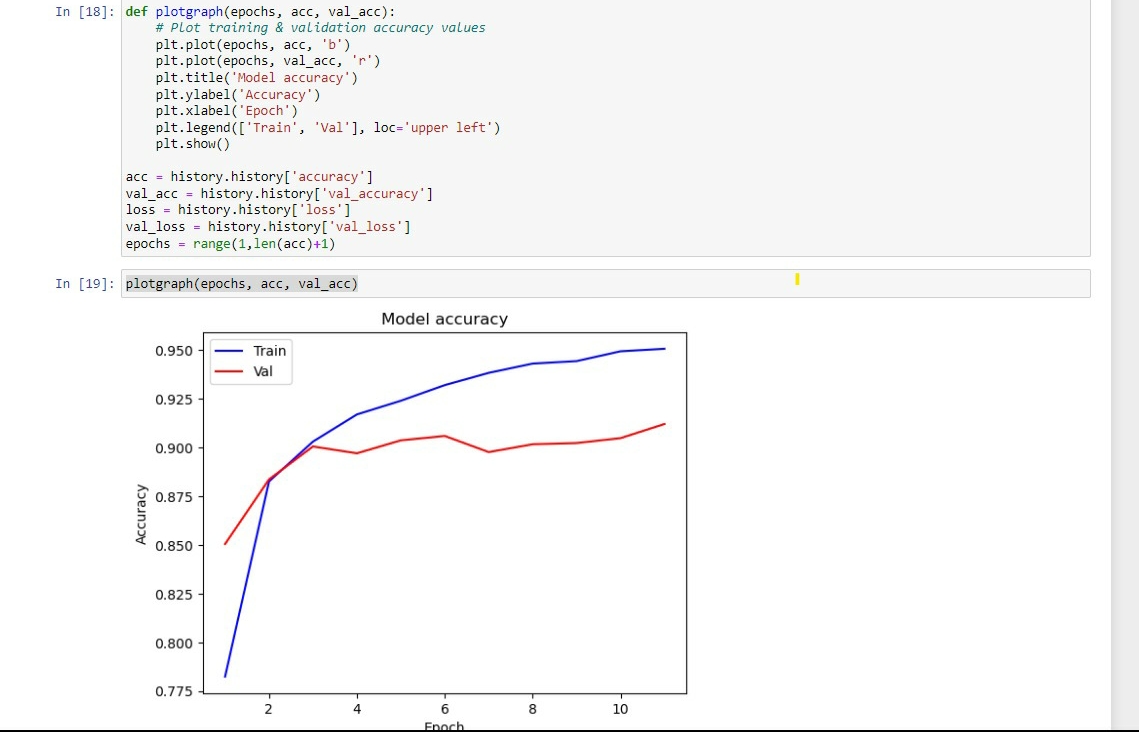
accuracy, save them in neural\_network3.h5 verbose is 1 to see the output progress bar while saving the weights.

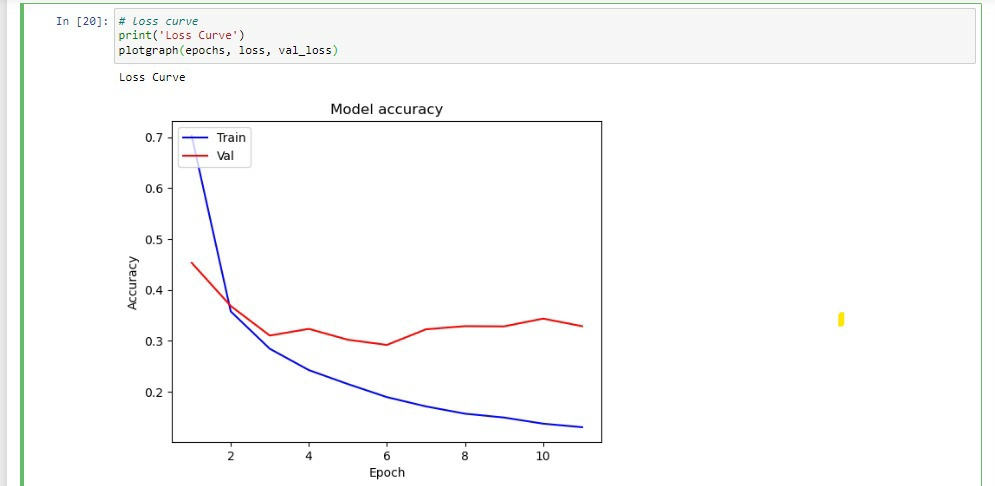


we store all the data about model training in history variable to visualization it using

matplotlib.pyplot library or we can use it to know any information about model training

like loss, val\_loss and accuracy.





Forth step: Save & load model:

We save the model after generating the weights and checkpoints, to use it at any time

without training from scratch. To load the model we load it into variable through

models.load\_model

Plots: we use plots to show the history of loss during training accourding to the epoch.

That show we have reached the minimum loss and save it.

Fifth step: Testing the model:

After that we resized it to (test\_images.shape[0],28,28) then we converted the image to np array. We use np.argmax to return the maximum possible value, this value is the input of the label\_dict to generate the correct letter.

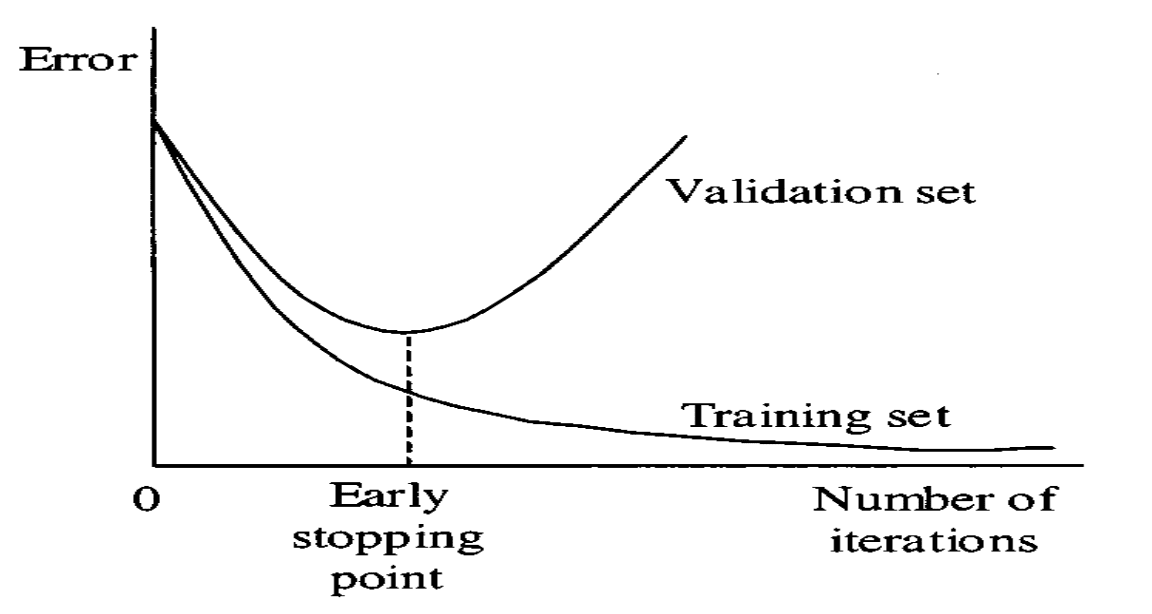


That was for the testing in coding then we continue testing our model with the active GUI.

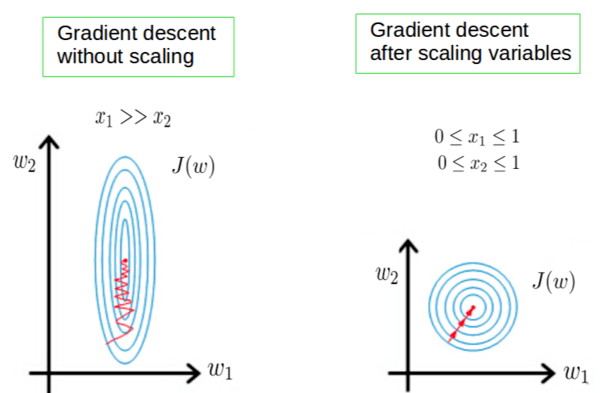
Model Evaluation:

We followed some steps in our model during the design to reach the best results.

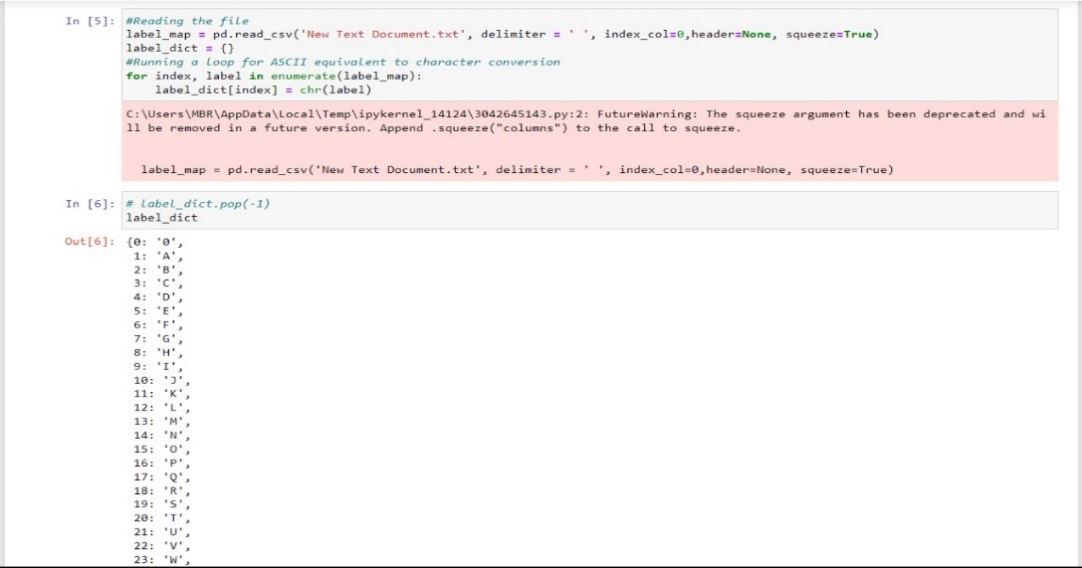
1. Regularization: with the early stopping, we added this feature to the model so it will stop training the model in a point which where there is no point to continue because this will result in a better accuracy.



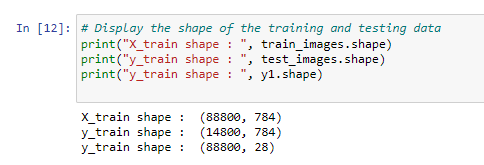
1. Rescaling: we rescaled all the images by dividing the training and the testing data by 255 this will make our machine learning model train faster with smaller images and result in saving time and resources



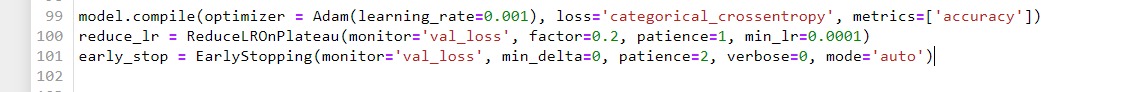
1. Converting: we converted the output which was numbers representing the letters into actual letters(Mapping) so we can recognize the output easily as users using (.txt) file.

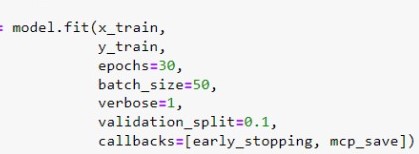


4.Reshaping: Reshaping basically means, changing the shape of an array. And the shape of an array is determined by the number of elements in each dimension. Reshaping allows us to add or remove dimensions in an array. We can also change the number of elements in each dimension, and we had to do it so the model can train well.



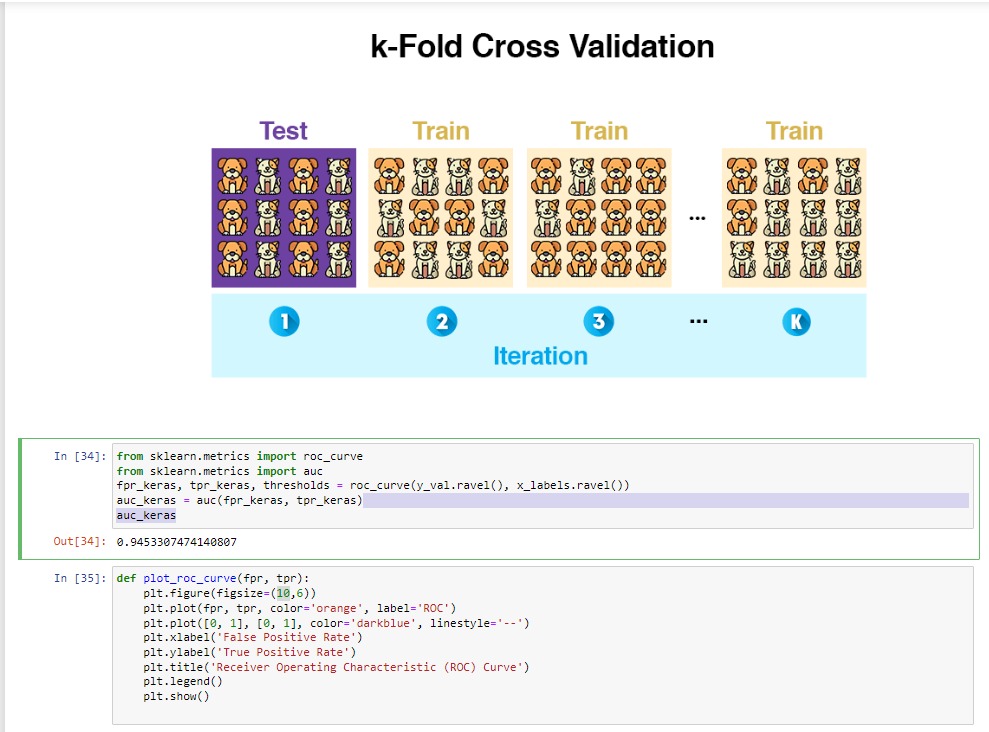
5.hyperparameter tuning: we tried our best in all the hyperparameters so we can get that accuracy and result.

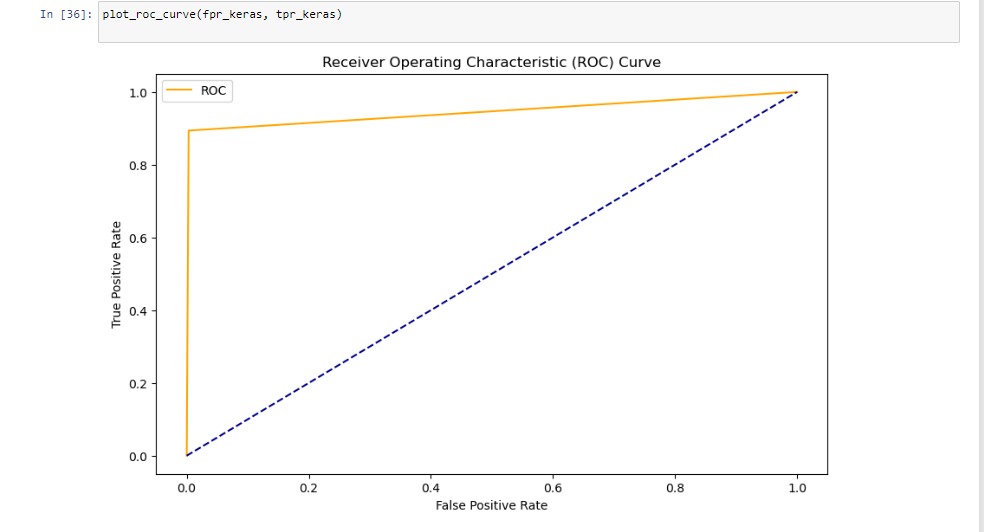




6.Visualization: we used many plots for better understanding to the model and to see all the progress.

7.K-fold: K-Fold is validation technique in which we split the data into k-subsets and the holdout method is repeated k-times where each of the k subsets are used as test set and other k-1 subsets are used for the training purpose.





Development platforms:

We used google Kaggle and Jupyter notebook to write and execute arbitrary python code.

• Numpy: is used to convert the image into an array and to deal with it.

.Sklearn: we used it for splitting the dataset.

• Keras: deep learning API written in Python, running on top of the machine learning platform

TensorFlow.

• Tensorflow: TensorFlow is the open-source library for several various tasks in machine learning.

• Matplotlib: Matplotlib is a comprehensive library for creating static and interactive visualizations in

Python to virtualize data.

• Tkinter: Tkinter is the standard GUI library for Python.

* Pandas: we used this library to read the csv file.



GUI sample:

