**HANDWRITTEN DIGIT RECOGNITION SYSTEM**

Anusha Dasari ([adasari@hawk.iit.edu](mailto:adasari@hawk.iit.edu)); Chirag Prakash ([cprakash@hawk.iit.edu](mailto:cprakash@hawk.iit.edu)); Vinjal Doshi ([vdoshi4@hawk.iit.edu](mailto:vdoshi4@hawk.iit.edu))

Code: https://github.com/AnushaDasari-DS/ML-Project--Spring-2021

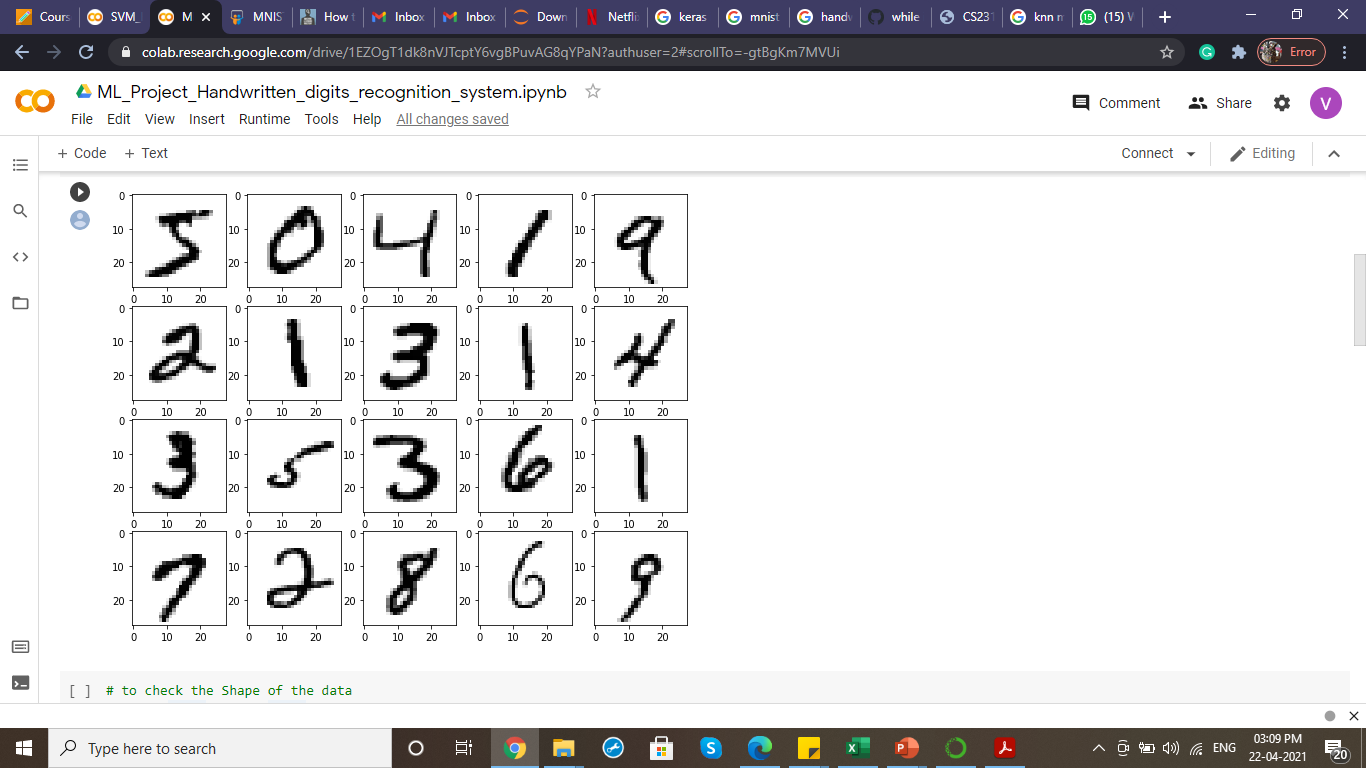
**Abstract**

*A handwritten digit recognition system using Logistic regression, Support vector Machine (SVM,) K-nearest neighbours (KNN) and Convolutional neural network (CNNs) is described in this report. MNIST dataset which consists of 60,000 images of handwritten images are given as input to the models, accuracy values are calculated using 10,000 test images and are compared against each model. Result says that CNN was able to achieve 99% of accuracy which is highest than any other model used. CNNs are very effective in automatic feature extraction that helps in recognizing the structure of handwritten digits and makes CNN the most suited way of solving handwritten recognition problems*.

**Introduction**

Handwritten digit recognition is one of the most fascinating and active area of research in pattern recognition in recent years. Now a days information retrieval from images has become popular and is being used widely in many applications. For example, while purchasing in online store, it will ask for a picture of our credit card. Using image recognition, it can identify the card number from the picture and make a payment. Similarly, pattern recognition can identify the handwritten digits written by humans. It is one of the challenging tasks to accurately identify the digit as the handwriting differs from person to person. Even though it easy for humans to recognize the handwritten digits, it is not for machines. We need to provide them with training data with digit images along with its label for machines to learn and identify. How the image represented as data is very important because the handwritten digit’s structure and features should be correctly represented in pixels.

In 1998, Modified National Institute of Standards and Technology (MNIST) has created an image database with a set of two NIST’s databases – Special Database 1 which consists of digits written by high school teachers and Special Database 3 which consists of digits written by United States Census Bureau employees. MNIST dataset has total of 70,000 images in which 60000 images are training set and 10000 images are of training set. The images in this dataset are of black and white which has resolution of 28\*28 pixels. This dataset also contains labels of each digits 0-9.



*Figure 1: Sample images of MINST Dataset.*

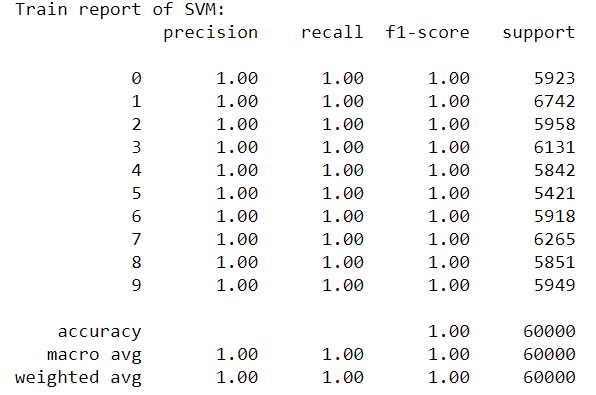
Sample digits from MNIST dataset is shown in Figure 1. Using neural network many researchers have achieved near human performance. From 1998, researchers have been trying many algorithms like Linear Classifier, K-nearest Neighbour, Non-Linear Classifiers and Neural networks. The test error was 12% with Linear Classifier in 1998 and it reduced to 0.23% with Convolution nets in 2012. In this project we are going to apply Logistic regression, Support Vector Machine, K-nearest Neighbour and Convolution Neural Networks to MINST dataset, calculate and compare the accuracies.

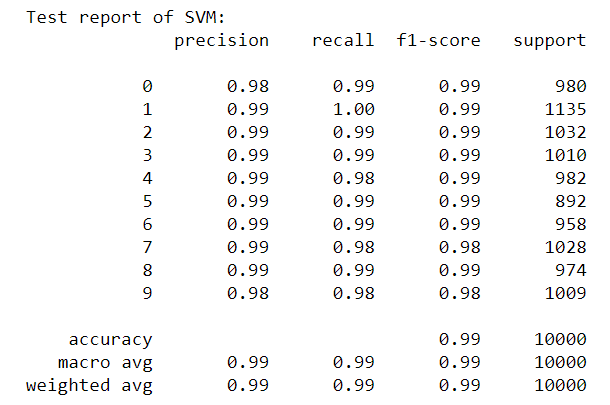
**Support Vector Machine (SVM)**

Support vector machine is a supervised learning algorithm. It can be very useful for both classification and regression. In this type of algorithm, we consider n-dimension space with data points as dots in that space. The algorithm looks for the most appropriate hyperplane with respect to those points. Regularization of the parameters is one of the advantages which allows to tackle overfitting in the algorithm.

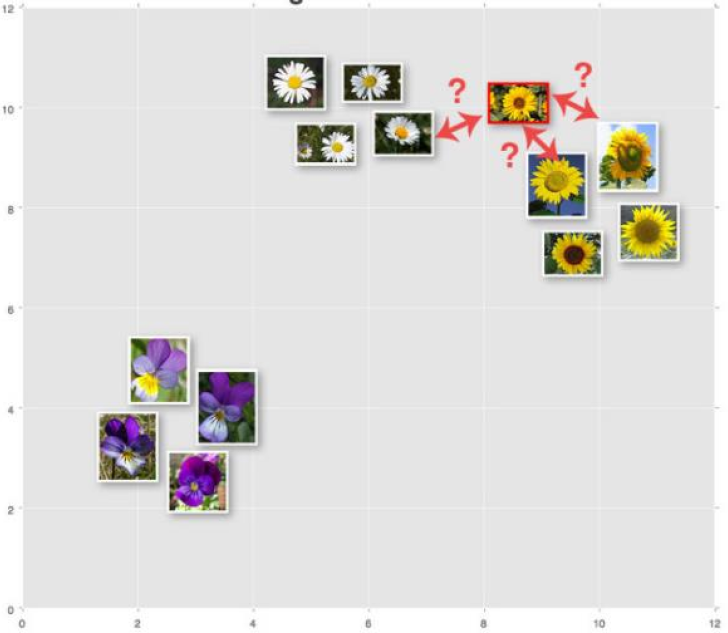


*Figure 2: Graph explaining SVM, Source:* [*https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/*](https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/)

The support vector machine in scikit-learn can take as input both dense along with sparse vectors as input. In scikitlearn have three classes that allows us to perform multiclass classification on a dataset which is SVC, NuSVC and LinearSVC. In this project we will use Linear Support Vector Classification class to perform the classification of MNIST dataset. LinearSVC or Linear Support Vector Classification uses a linear kernel and adapts in terms of a liblinear that has more flexibility in the selection of penalties and loss functions and should scale better large numbers of samples. Below is the accuracy for the SVM algorithm.

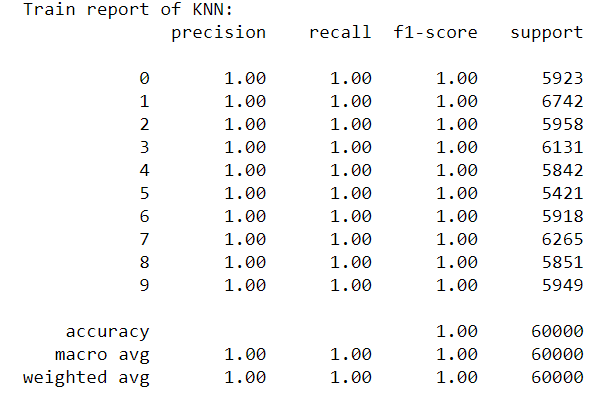


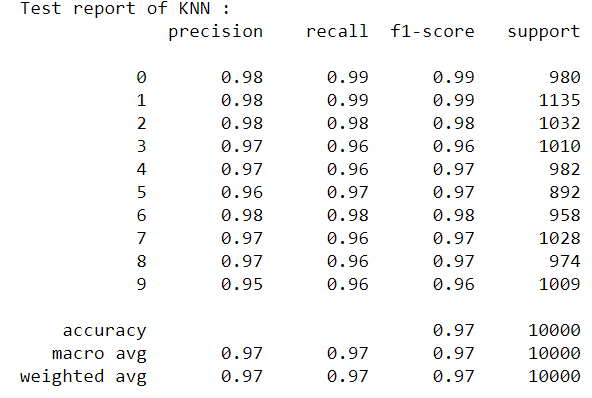
**K-Nearest Neighbour (KNN)**

KNN classifier is the most simple image classification algorithm that can be used for categorical problems. KNN algorithm depends on the distance between feature vectors. KNN algorithm categorizes input data by locating the most common class among the k closest data points. Each data point in the k closest votes and the largest category number of votes wins. 

*Figure 3: Example of KNN, Source:* *https://customers.pyimagesearch.com/lesson-sample-k-nearest-neighbor-classification/*

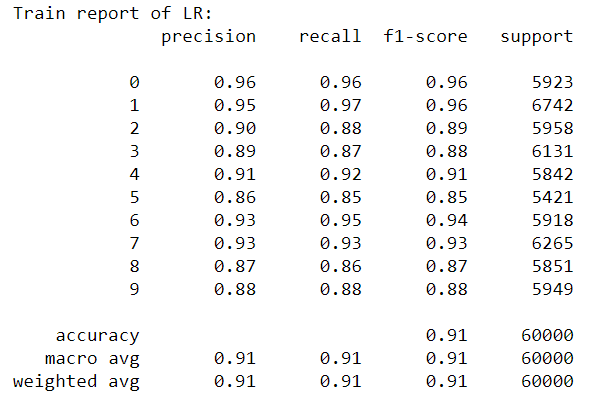
The benefits of using the KNN algorithm, that is, it is effective at noisy training data and it is very efficient if the data is very large sized. KNN algorithm requires training datasets which include perfectly labelled points. KNN is also a non-parametric classifier. The algorithm considers new data points as its input and performs classification by calculating distance between new and labelled data points using the Euclidean or Hamming distance. Below is the accuracy for the KNN algorithm.

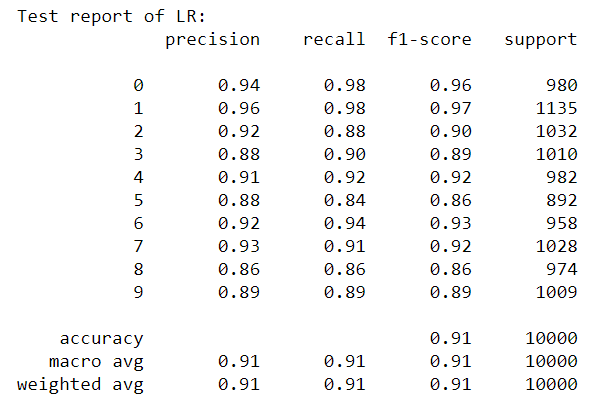




**Logistic Regression**

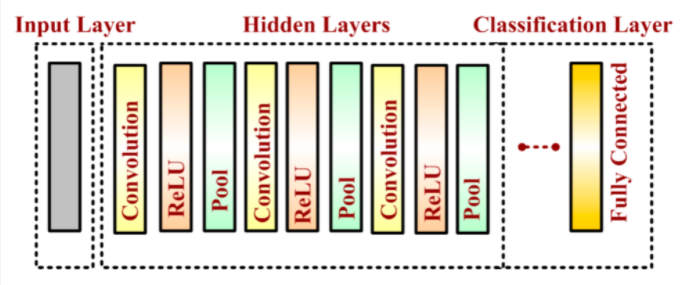
Logistic regression is an algorithm which uses logistic function to predict a binary dependent variable. In regression analysis, logistic regression is estimating the binary classification of a logistic model. Logistic regression is the efficient regression analysis to conduct when the dependent variable is binary. The logistic regression does analysis as a predictive algorithm. Logistic regression is used to relate data and to define the relationship between one dependent binary variable and one or more ordinal variables. Below is the accuracy for the logistic regression





**Convolutional Neural Network (CNN)**

CNN is one of the most popular models used for image classification. It was first presented by Yann LeCun in 1998 for digit classification. CNN is a special form of multi-layered feed forward neural network which was inspired from biological multilayer perceptron neural network (MLP) designed to emulate human visual cortex. This model uses adjacent pixel information to create the features by down sampling the image and use prediction layers. It consists of 3 components – convolutional layer, pooling layer and output layer as shown in Figure 2. For image classification, a three convolutional layer is well adapted which is of input layer, hidden layer that can consists of multiple convolutions, activations and pooling and a classification layer in which fully connected to output layer.



*Figure 4. Convolutional neural network architecture*

*Source:* [*https://www.mdpi.com/1424-8220/20/12/3344/html*](https://www.mdpi.com/1424-8220/20/12/3344/html)

Explanation of each layer in CNN:

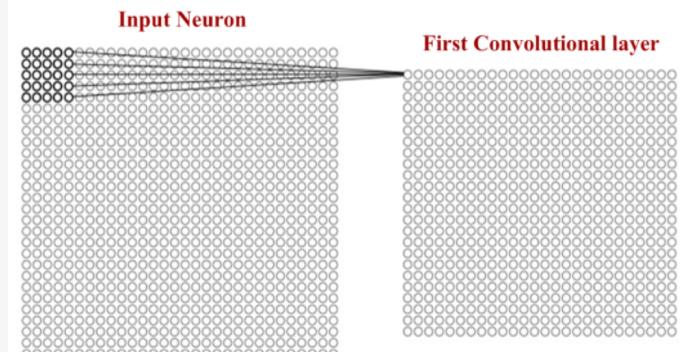
*Input Layer*:

This layer consists of images which are passed as input to the model.

*Hidden Layer:*

This layer consists of series of convolution, activation, and pooling functions. With these functions features of image are extracted.

Convolution layer: The input image goes through multiple convolution filters or kernels, where each filter extracts a feature by computing dot product. The (n x n ) part of image is convoluted by (m x m) filter giving output as (n-m+1) \*(n-m+1) matrix. This output saved and we slide the kernel by 1 pixel. We choose the value by which the kernel should stride and extract different features as shown in Figure 3. If we want to keep the feature size same as input image size, we add zeros around margin to increase its dimension and this process is known as padding.



*Figure 5. Filter of 5x5 with activation map, Source:* [*https://www.mdpi.com/1424-8220/20/12/3344/htm*](https://www.mdpi.com/1424-8220/20/12/3344/htm)

The kernel values are learned by neural network using backpropagation.

Activation layer: Like other neural networks, CNN uses activation functions like sigmoid function, rectified linear unit (ReLu) etc to get nonlinearity in the system which allows CNN to have ability to learn non-linear input functions. ReLu is proved to be efficient than other activation functions in CNN models.

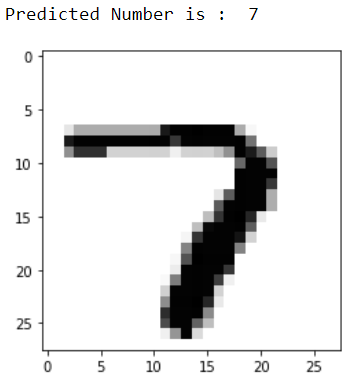
Pooling layer: Pooling layer helps to subsample the featured map by reducing spatial dimensionality and helps in reducing overfitting of the data by giving abstract representation. It acts just like convolution layer, but pooling takes maximum of the region instead of calculating dot product between kernel and input pixel points.

These steps can be repeated multiple times as required and is passed to fully connected neural network which is added at the end of CNN to help the model learn non-linear features as outputs by convolution layers.

For MINST dataset last layer will be of size 10 as each node of a digit which is output of softmax activation function in fully connected layer gives output as 10-dimensional vector with probabilities for each digit. Below is the accuracy of CNN model:

313/313 [==============================] - 1s 4ms/step - loss: 0.0217 - accuracy: 0.9930

Output:



**Graphical User Interface (GUI) implementation using OpenCV**

GUI helps us to make the system more user friendly and creates a creative impact on the user. GUI is responsible for adding a visual component to our software. User interface plays a very important phase in the deployment of the software project.

OpenCV: It is an open-source computer vision library in python. It has various function through which a user can interact with our software, there are various function which can help us in accessing the cursor, providing the space for drawing.

OpenCV's application areas include:

* 2D and 3D feature toolkits
* Facial recognition system
* Gesture recognition
* Object detection
* Segmentation and recognition
* Motion tracking
* Augmented reality

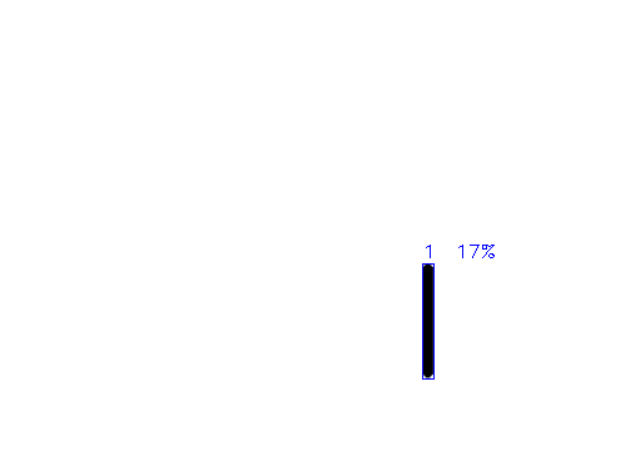
In the user interface we created,

Canvas: This is basically the space where the user can draw the handwritten digit and check.

Buttons: There are two buttons at the bottom of canvas one which clears the handwritten digit drawing and one that detects/recognise the digit from the user drawing.



*Figure 6: Canvas to draw a digit*



*Figure 7: Output with digit recognized*

**Results**

We have implemented four different models on our data to come up with the best model that fits perfectly with our data, and the one which shows more accuracy and less error percentage overall. After implementation we have received the following results:  
First model we implemented was:

|  |  |  |
| --- | --- | --- |
| **Algorithm** | **Train Accuracy** | **Test Accuracy** |
| Logistic Regression | 91% | 91% |
| K nearest neighbour | 100% | 97% |
| Simple vector classifier | 100% | 99% |
| CNN | 99.30% | 99.30% |

*Table 1: Accuracy comparison on models*

**Future Work**

We can create an application with a more better user interface. We can create our CNN model with a greater number of hidden layers. While training the model we can split the data in various ratios to better train our model. Since CNN model has shown most accurate results, we can create the larger convolutional network.

The system can be used in various application such as postal mail sorting, bank check processing, form data entry, etc. One can also create an application using dash in python to create a better user accessible software.

**Conclusion:**

In this project, we have implemented Handwritten Digit Recognition using various widely used Machine learning models like SVM, Logistic regression, KNN and CNN on MINST dataset which contains images of handwritten digits. We have evaluated each model and have conclude that the convolutional neural network works most efficiently with accuracy of 99.30% followed by support vector classifier with accuracy 99%, while KNN and Logistic Regression are not that effective.

**Contributions:**

Below are contributions from team members on this project:

Anusha Dasari – Have worked on implementing CNN model on MINST dataset.

Chirag Prakash – Have worked on implementing SVM and KNN on MINST dataset.

Vinjal Doshi – Have worked on implementing Logistic regression model on MINST dataset.

Everyone has contributed equally on creating GUI and on creating this final project report.

**References**:

* <https://towardsdatascience.com/mnist-handwritten-digits-classification-using-a-convolutional-neural-network-cnn-af5fafbc35e9>
* <https://www.mdpi.com/1424-8220/20/12/3344/htm>
* <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.160.8494&rep=rep1&type=pdf>
* <https://en.wikipedia.org/wiki/MNIST_database>
* <https://en.wikipedia.org/wiki/Handwriting_recognition>
* <http://users.eecs.northwestern.edu/~mvb541/EECS349/Report.pdf>
* <https://ijisrt.com/wp-content/uploads/2019/06/IJISRT19JU358.pdf>
* <https://towardsdatascience.com/convolution-neural-networks-a-beginners-guide-implementing-a-mnist-hand-written-digit-8aa60330d022>
* <https://ieeexplore.ieee.org/abstract/document/8239052?casa_token=1BjxKDc0gnkAAAAA:Sr_Aq-kgd5h-DD6Qu1G9s3uemWbhTfNtrRsfkC6r2HCAMk3bTX9mibc6ME5to1Vou6uC3Q>
* <https://arxiv.org/abs/1811.08278>
* <https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/>
* [*https://customers.pyimagesearch.com/lesson-sample-k-nearest-neighbor-classification/*](https://customers.pyimagesearch.com/lesson-sample-k-nearest-neighbor-classification/)