Introduction

Our project is about recognizing handwritten images of alphabets and solving handwritten math equations which include basic operations such as addition, multiplication and subtraction and displaying the prediction for the same on a web app which is deployed on Flask.

Handwritten digit recognition is one of the classical problems in deep learning. It is a challenging task since handwriting differs from person to person. The digits written by different people may not be of same size, thickness, or orientation and pressure applied while writing. Although, this difference in handwriting does not cause any problems to humans. However, it is more difficult to teach computers to recognize general handwriting.

Basic approach we followed is the classification of images using CNN (convolutional neural network). CNN is primarily used in object recognition by taking images as input and then classifying them in a certain category. Handwritten character recognition is one of that kind. We will be having a set of images which are handwritten character with labels of each type. CNN the most suitable approach for solving handwriting recognition problems.

We will be discussing the code which we used to train our data and designing model from it. Further discussions include the algorithm used and demonstration of the same on a web app.

Problem Definition and Algorithm

2.1 Task Definition

Task is to predict the alphabets and the result of mathematical equations with maximum accuracy which is provided by user in the form of images to the model.

The webapp gives user the choice to select between alphabets and equations and accordingly user is required to provide the image. This serves as input to the model.

The prediction is displayed on the screen.

This becomes an interesting problem to solve as handwriting differ from person to person, in terms of style and size, its difficult to get it recognized by computers so our model serves the purpose by recognising characters with accuracy of around 97- 99%.

2.2 Algorithm Definition

The algorithm used to perform the task stated above is CNN. CNNs are very effective in perceiving the structure of handwritten characters/words in ways that help in automatic extraction of distinct features. A CNN can extract affluent and interrelated features automatically from images.

Input data -> Pre-processing -> Segmentation -> Feature generation -> Classification -> Recognition.

- Input data is a dataset of all characters.
- Next step is preprocessing, where an image is subjected to various operations like noise reduction, document skew correction, slant correction, normalization, smoothing, and skeletonization.
- Segmentation of an image is done to isolate the characters of an image into different sub-images. Each sub-image is considered as one individual character.
- The next phase is feature generating, in which various extraction techniques are used to represent an image as a vector feature in the feature generator.
- The feature generation is followed by classification.
- Since handwritten digits recognition contains a wide variety of options the way the digits are written, the main concern of this study is to recognize handwritten digits.

Experimental Evaluation

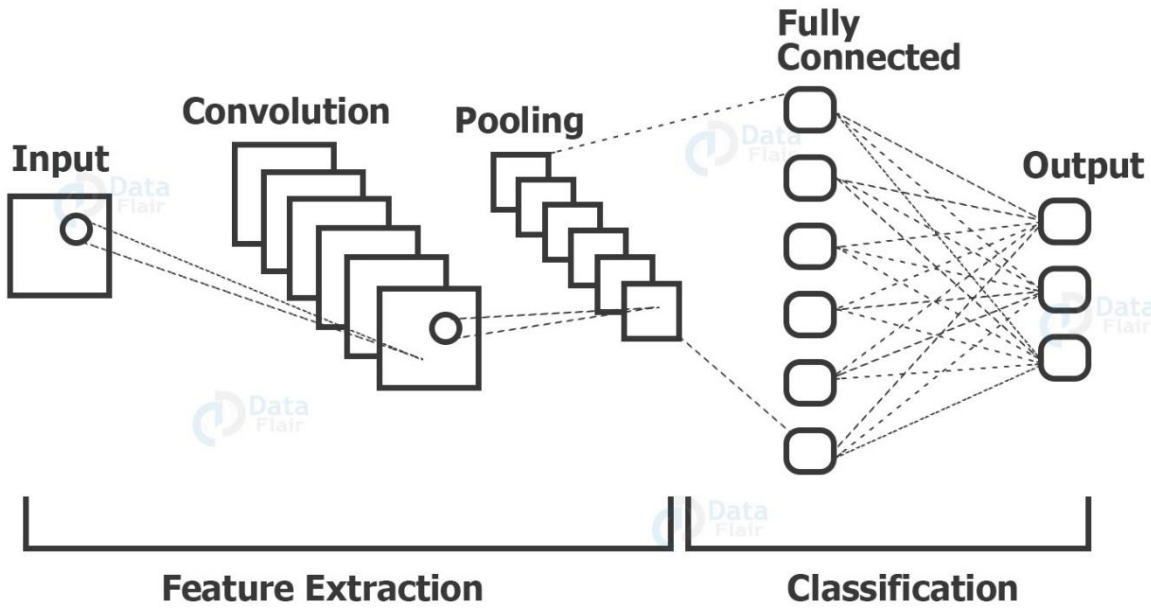
3.1 Methodology

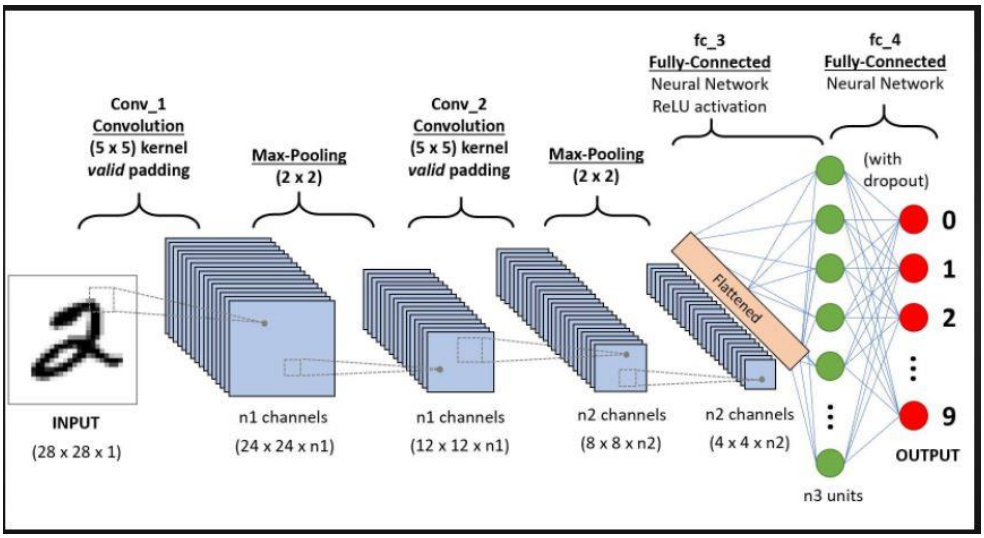
Initially, we split the dataset into training and testing dataset. We are reshaping the train & test image data so that they can be displayed as an image, as initially in the CSV file they were present as 784 columns of pixel data. So, we convert it to 28×28 pixels.

```
In [6]: train_x, test_x, train_y, test_y = train_test_split(X, y, test_size = 0.2)
        train_x = np.reshape(train_x.values, (train_x.shape[0], 28,28))
        test_x = np.reshape(test_x.values, (test_x.shape[0], 28,28))
        print("Train data shape: ", train_x.shape)
        print("Test data shape: ", test_x.shape)

        Train data shape:  (297629, 28, 28)
        Test data shape:  (74408, 28, 28)
```

We evaluate the model by reshaping of training and testing dataset followed by model creation (CNN in our case). The basic workflow of how CNN works -

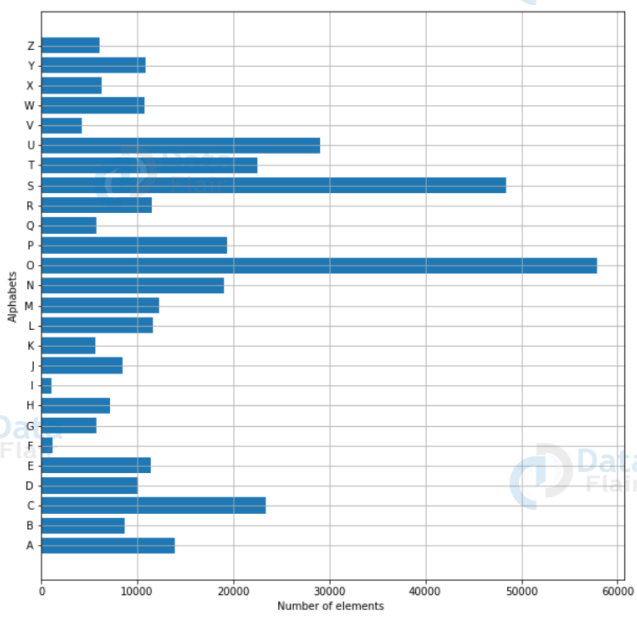




Once the model is created, we compile the model and fit training dataset. Validation accuracy and losses are computed followed by prediction on testing dataset.

Architecture	Training accuracy	Testing accuracy	Real time accuracy
MLP	0.98	0.92	0.62
CNN	0.98	0.98	0.84

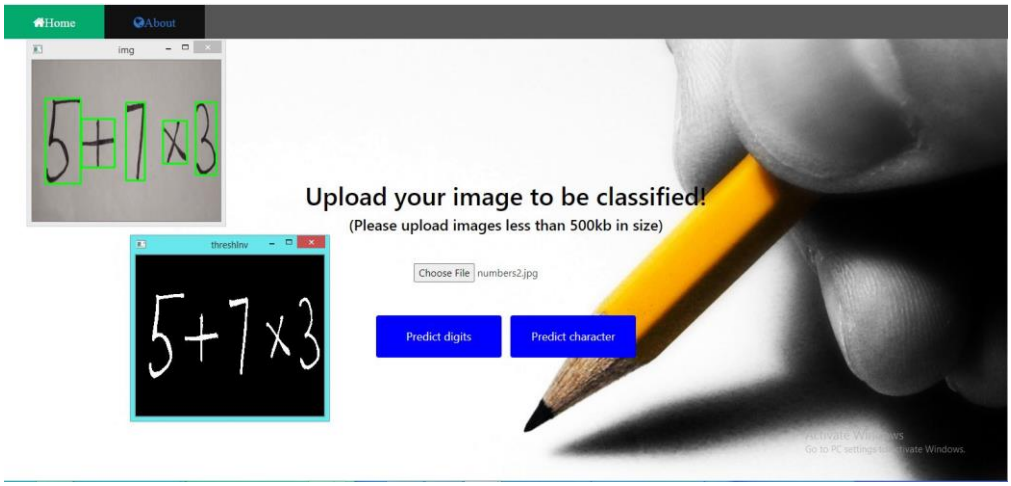
3.2 Results

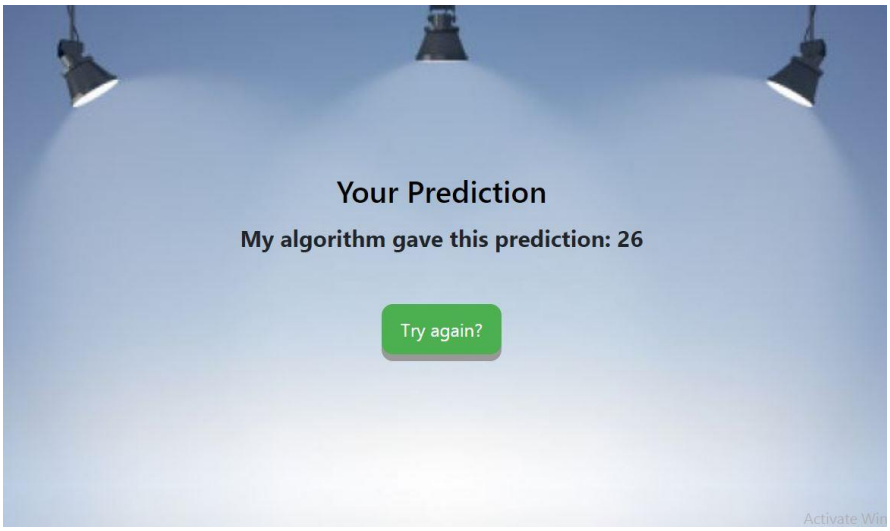


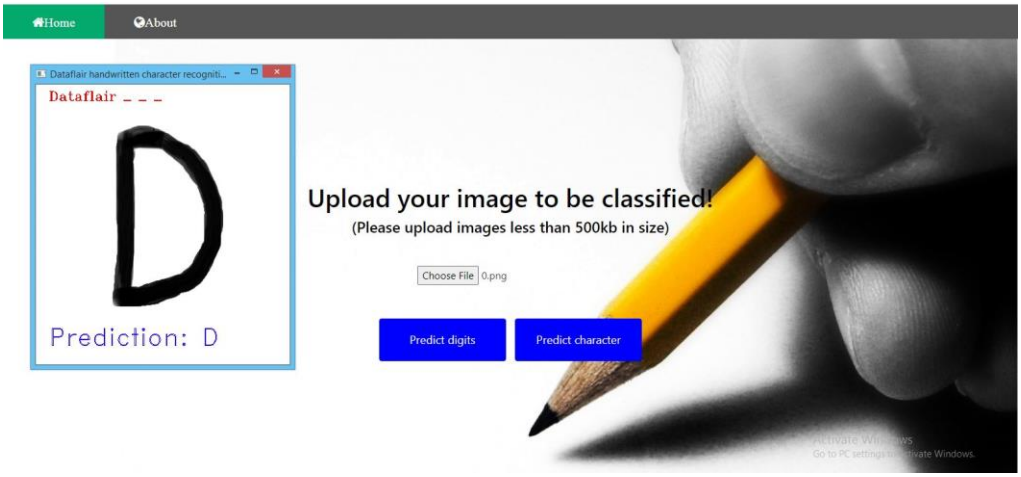
In [7]:

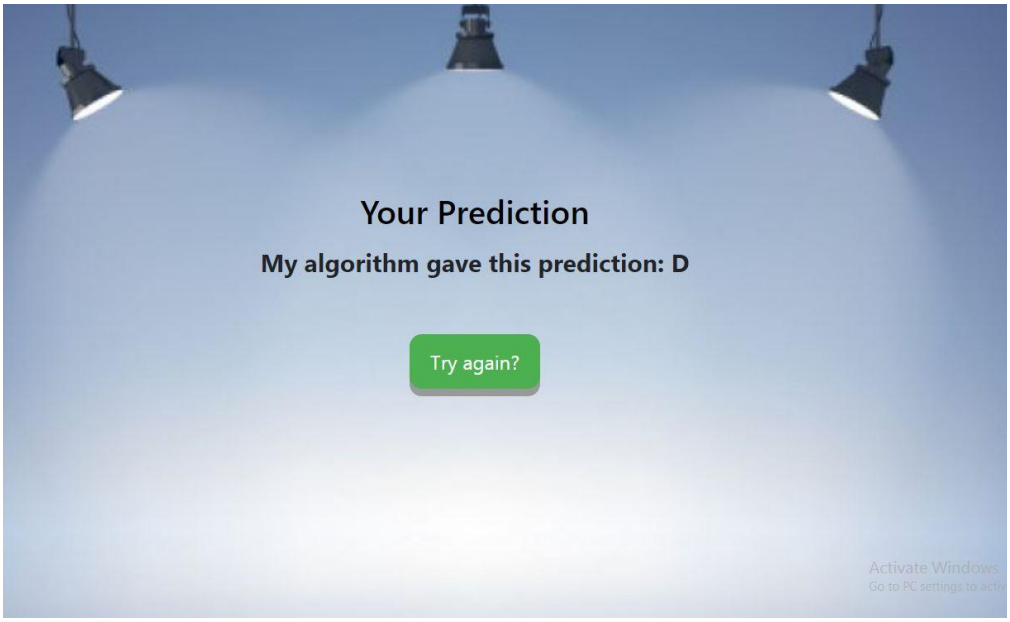
```
train_gen = gen.flow_from_directory(folder_path,
                                    target_size = (32,32),
                                    color_mode = 'grayscale',
                                    class_mode = 'categorical',
                                    batch_size= 64,
                                    shuffle=True)
```

```
Found 47827 images belonging to 14 classes.
```









3.3 Discussion

- Since the model is based on prediction, even though after achieving accuracy of around 97 to 99 %, we do not get the expected results always.
- Trying with our own custom dataset resulted it very less accuracy.
- It is really a challenging issue to develop a practical hand- written character recognition (CR) system which can maintain high recognition accuracy. In most of the existing systems recognition accuracy is heavily dependent on the quality of the input document.

Related Work

Techniques used till now and their disadvantages are discussed in the following section.

1. **OCR (optical character recognition)**

OCR is used to extract important data from document, it helps in differentiating presence of characters by their shapes. In OCR, scanner is provided with character recognition software which converts bitmap images of characters to equivalent ASCII codes.

First step in whole process is to create bitmap of image of document then with help of software OCR translates thee array of grid points into ASCII text which pc can understand and process it as letters, numbers and special characters.

OCR software process bitmap of every character and compares it with set of characters which machine has been programmed to acknowledge to translate bitmaps into text, Whichever character pattern it matches or nearly matches, is taken into account to be character read. If a scanned character doesn't match with any of the already stored character patterns, it's rejected.

Disadvantages:

- OCR text works efficiently with the printed text only and not with handwritten text. Handwriting must be learnt by the pc.
- OCR systems are expensive. There is the need of lot of space required by the image produced.
- Not 100% accurate, there are likely to be some mistakes made during the method. Not worth doing for little amounts of text.

2. **KNN Algorithm**

K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.

Disadvantages-

- It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.

- Accuracy depends on the quality of the data.
- With large data, the prediction stage might be slow.
- Sensitive to the scale of the data and irrelevant features.
- Require high memory – need to store all of the training data.

3. **SVM Algorithm**

   In the SVM algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiates the two classes very well.

   Disadvantages-

- SVM algorithm is not suitable for large data sets.
- SVM does not perform very well when the data set has more noise i.e. target classes are overlapping.
- In cases where the number of features for each data point exceeds the number of training data samples, the SVM will underperform.
- As the support vector classifier works by putting data points, above and below the classifying hyperplane there is no probabilistic explanation for the classification.

  Disadvantages reflected by above mentioned techniques lead to use of advanced technology which is efficient as well as easy to implement. Use of CNN gave extra ordinary performance accuracy with very less disadvantages and overcome the shortcomings of most of the listed technologies.

  CNN algorithm achieves the best performance and the best classification result, the accuracy is up to 99%.

Future Work

The future work would be mostly the extension of current project which will include recognising statements from the text image and predicting it through a voice enabled feature as currently the predictions are just displayed on the screen.

We would also try to solve more complex equations and add other functionalities such as calculating square root, logarithm of given number etc. as currently its limited to only four basic operations.

We would also try to extend it to other languages as currently its just limited to English alphabets.

Conclusion

We have successfully developed Handwritten character recognition (alphabets and solving mathematical equations) with Python, Tensor flow, keras and Machine Learning libraries.

Handwritten characters have been recognized with more than 97% test accuracy. This can be also further extended to identifying the handwritten characters of other languages too.

Predicting results of the handwritten equations have achieved accuracy of up to 99 %. This can be further extended to solve complex equations.

Bibliography

- https://link.springer.com/chapter/10.1007/978-981-13-1651-7_47
- https://www.mdpi.com/1424-8220/20/12/3344/htm