Udacity Machine Learning Engineer Nanodegree

Capstone Project

Handwritten Digits Recognition Web App

Abdelrahman Fekry Ali

8/28/2021

Domain Background

in nowadays the need for Document digitization is in demand. due to many benefits such as the ease of accessing the data in the digital form from anywhere and cost reduction and data security through defining accessibility privileges and data storage and recovery.
in computer vision field the process of converting these documents into digital form is called Optical Character Recognition (OCR).
the process of Optical Character Recognition (OCR) includes many steps, in this project we are going to focus on the base step which is Handwritten digit recognition (HDR).

Handwritten digit recognition (HDR) is considered a base step for many applications such as

- Signature Verification,
- Postal-Address Interpretation,
- Bank-Check Processing,
- Writer Recognition.

Problem Statement

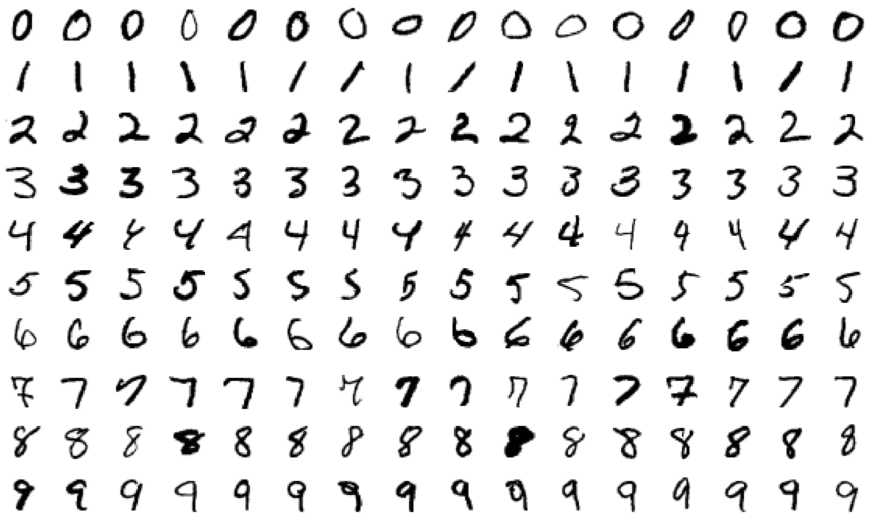
the main goal of this project is to implement web app that as asks the user to sketch a digit and then send the pixels information to backend where the data is processed then return the predicted value to the frontend.

Datasets and Inputs

for this project we are going to the MNIST (Modified National Institute of Standards and Technology) dataset.

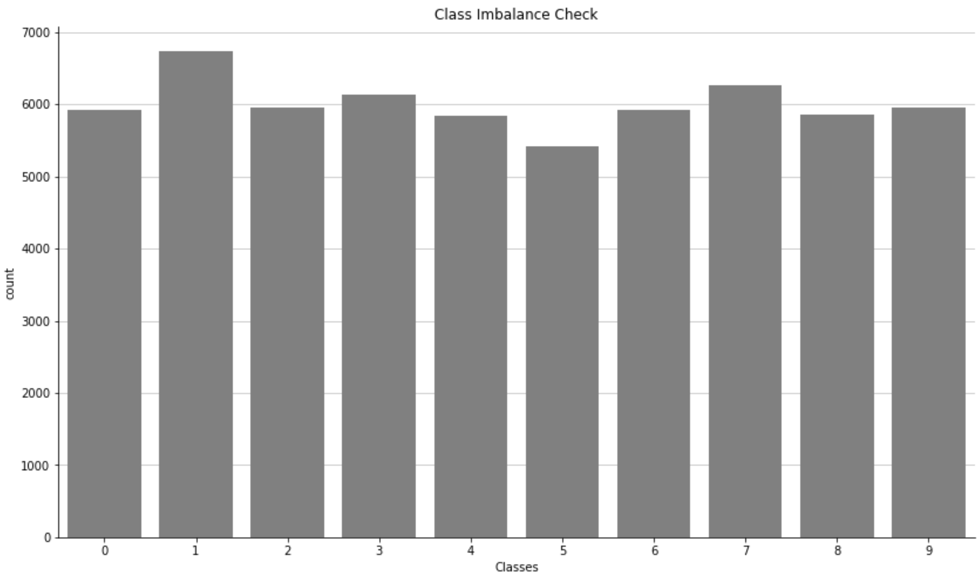
base on the paper[1] The set of images in the MNIST database was created in 1998 as a combination of two of NIST's databases Special Database 1 (digits written by high school students) and Special Database 3 (digits written by employees) of the United States Census Bureau.

There are 70,000 images and each image has 784 features. This is because each image is 28 x 28 pixels, and each feature represents a pixel’s intensity, from 0 to 255.



Sample images from MNIST test dataset

The dataset doesn’t have missing values. The Classes are approximately balanced as shown below



Class Imbalance Check

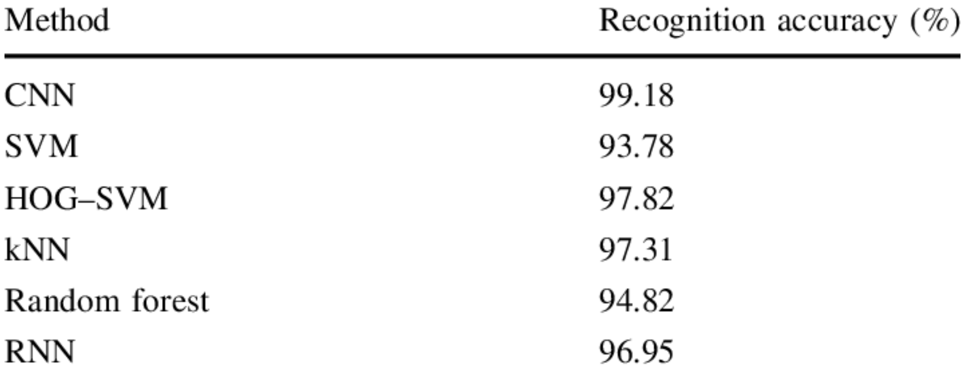
Solution Statement

The solution includes 3 main stages:

- Structuring, Training and saving model as pre-trained model.
- Designing and constructing the user interface "FrontEnd" and its elements such as the drawing canvas, reset button and submit button.
- Constructing the BackEnd where Canvas pixels are received and processed in the pre-trained model.
- Returning the Predicted Value from the BackEnd and rendering it on user interface.

Benchmark Model

based on the paper[2] as comparing Recognition accuracy of machine learning methods on MNIST dataset the convolution neural network have the best accuracy which is the targeted metric for this classification problem



Method	Recognition accuracy (%)
CNN	99.18
SVM	93.78
HOG–SVM	97.82
kNN	97.31
Random forest	94.82
RNN	96.95

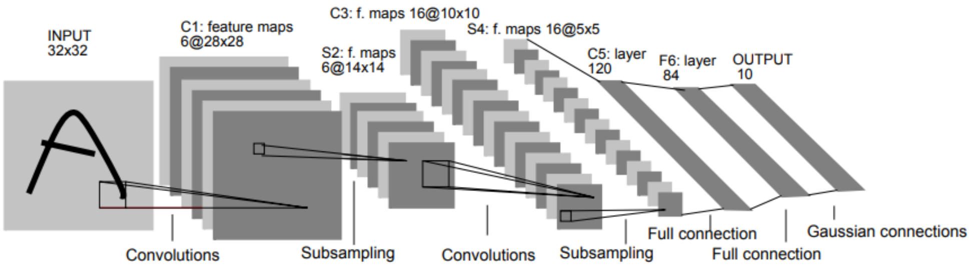
Recognition accuracy of machine learning methods on MNIST dataset

Evaluation Metrics

as dealing with classification problem that equally cares about Type I Error and Type II Error and labels Classes are balanced we are going to select accuracy as the evaluation metric

Project Design

- **Data Preprocessing**
    1. Missing Values Check
       check for any missing values in the data.
    2. Max-Abs Scaling
       scaling images pixels to be within the range (0-1).
    3. Adjusting Shape And Dimensions
       reshape the input to fit the model required dimensions.
    4. Class Imbalance Check
       check for class imbalance in labels and dealing with it if any.
    5. One-Hot Encoding
       One-Hot encoding the labels to fit the model required labeling.
    6. Data augmentation
       for more generalization we’re going to perform rotation, zooming and shifting on training dataset
- **Modeling**
    1. Structuring
       based on the paper[3] LeNet-5 CNN is considered as ideal for digit/character recognition so we are going to design our model base on it, with slight modifications.



Architecture of LeNet-5

-
    2. Training
       training CNN model.
    3. Evaluating
       evaluate the performance of the model based on the targeted metric (accuracy)
    4. Saving
       saving the evaluated model to load it in the backend of our application
- **FrontEnd**
    1. Canvas
       a blank space for use to draw on.
    2. Reset Button
       clearing the canvas.
    3. Predict Button
       collecting canvas pixels data and send it to backend to be processed and return the predicted value
    4. Text Area
       renders a message with predicted value
- **BackEnd**
    1. We’re going to use Flask for the backend as it written in python, we’re going to set the route for page rendering and a request/response route where process of collecting, processing and resending is occurred.

References

- [1] "THE MNIST DATABASE of handwritten digits" Yann LeCun, Corinna Cortes, Christopher J.C. Burges http://yann.lecun.com/exdb/mnist/
- [2] "ARDIS: a Swedish historical handwritten digit dataset" Huseyin Kusetogullari, Amir Yavariabdi, Abbas Cheddad March 2019 https://link.springer.com/article/10.1007/s00521-019-04163-3
- [3] "GradientBased Learning Applied to Document Recognition" Yann LeCun, Leon Bottou, Yoshua Bengio and Patrick Haffner ,November 1998 http://vision.stanford.edu/cs598_spring07/papers/Lecun98.pdf