**Indian Institute of Technology Indore**

**Discipline of Computer Science and Engineering**

**Minor Project in the course “Computational Intelligence”**

**Spring 2022-2023**

**Title: Text Detection in Images**

**Final Report**

**Team Members:**

|  |  |  |
| --- | --- | --- |
| Bhore Parth Shirish | 200001015 |  |
| Nishchay Shroff | 200001055 |  |
| Vipul Mahajan | 200001080 |  |

**Under the Supervision of**

**Dr. Aruna Tiwari**

**Professor, CSE**

**Problem Statement:**

Detecting text from images involves using image processing techniques and optical character recognition (OCR) technology to extract text from images and convert it into machine-readable format. The process typically includes pre-processing the image to improve the quality and contrast, detecting regions of interest that contain text, and using OCR algorithms to recognize and extract the text.

1. **Dataset Used**

The dataset used is called Standard OCR Dataset. It contains 45,500 images. These images are divided into A-Z and 0-9 characters, with each character having around 1200 images. This dataset will help the model to recognize different alpha-numeric characters.

Dataset Link: <https://www.kaggle.com/datasets/preatcher/standard-ocr-dataset>

1. **Data Pre-processing**

The 7 steps used for Image Data Pre-processing in Text Extraction for images are as follows:

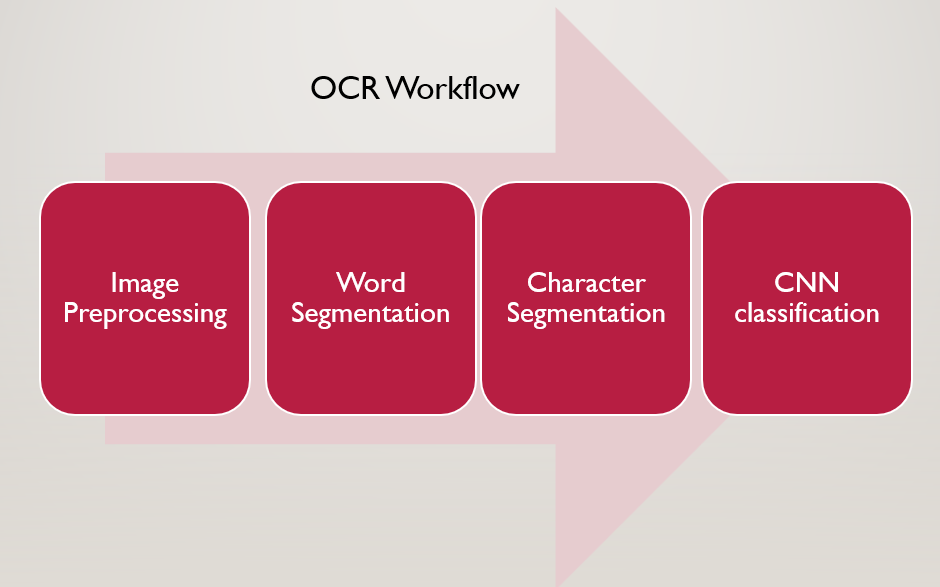
* Normalization: This process changes the range of pixel intensity values. It changes pixel range from [0, 255] to [0, 1]
* Skew Correction: The image might be slightly skewed or rotated. So, the image is de-skewed.
* Scaling: Scaling is used to increase pixel intensity. For character recognition, it should be more than 300 PPI (Pixels per inch)
* Noise Removal: This step removes the small dots/patches which have high intensity compared to the rest of the image for smoothening of the image.
* Thinning and Skeletonization: This step is performed for the handwritten text, as different writers use different stroke widths to write. It makes the width of strokes uniform.
* Gray-Scaling: This process converts an image from other color spaces to shades of Gray. The color varies between complete black and complete white.
* Thresholding or Binarization: This step converts any image into a binary image that contains only two pixel values determined by a threshold.

Image before pre-processing: Image after pre-processing:

1. **Algorithm**

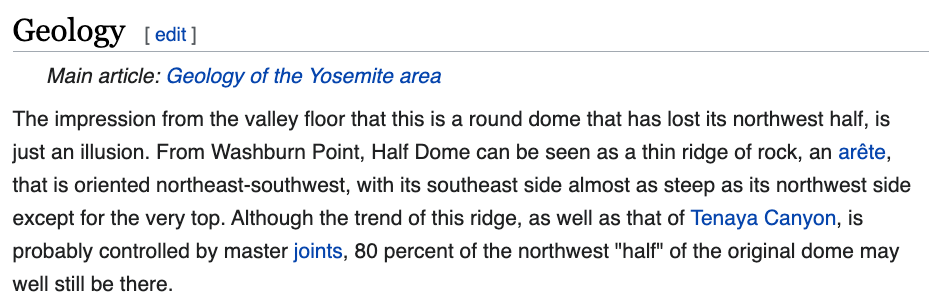
The OCR workflow is as follows:



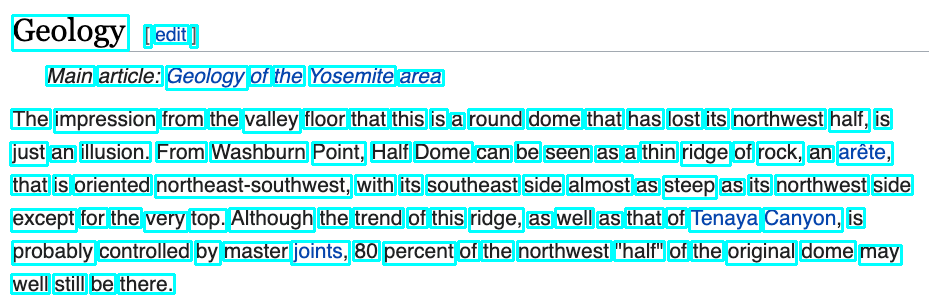
1. Word Segmentation:

* After Data-Preprocessing, Words are detected from the image using Contours.
* Change in Contour is used for Edge-Detection
* A bounding box is created across every word using the detected Edges

**Input:**

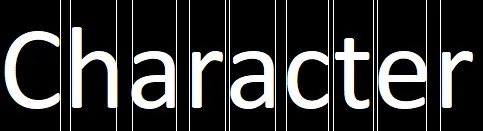


**Output:**



1. Character Segmentation:

* Vertical Projection Profile(VPP) is used to segment letters
* VPP calculates column sum of pixel values in the image
* Sum = 0 indicates break between characters. This is used to separate characters in a word



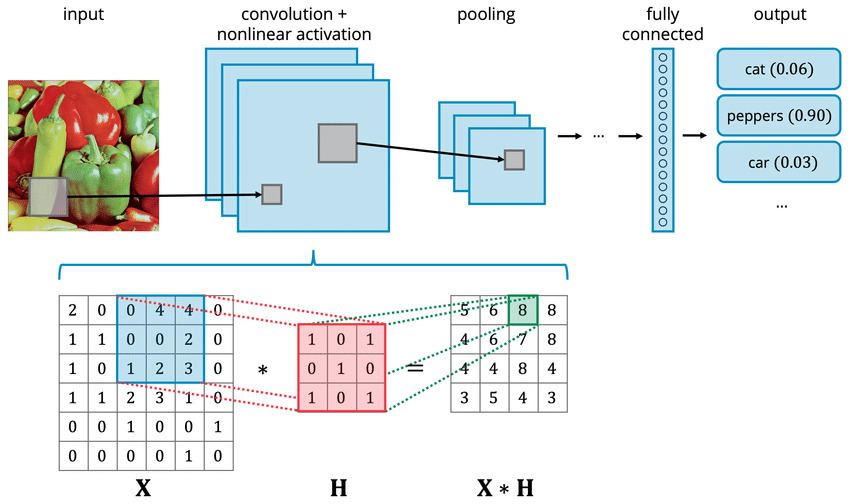
1. CNN Classification:

For Character Recognition, we plan to use Deep Learning with Convolutional Neural Network (CNN) architecture

The different layers used in CNN architecture are:

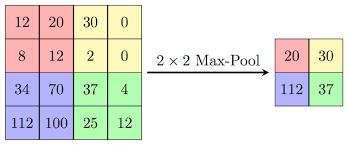
* Convolutional Layer:

1. Convolutional layer helps in feature extraction
2. Weight matrix called Kernel is multiplied with image to get features.



* Pooling Layer:

1. Pooling layer reduces size of feature map
2. Different types include Max-Pooling, Average-Pooling, etc.

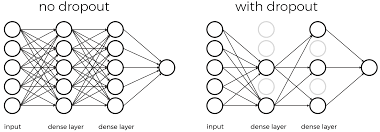


* Dense Layer

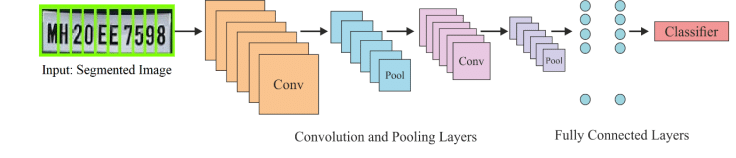
1. It is fully connected layer
2. Every neuron between 2 adjacent layers has a connection

* Dropout Layer

To prevent overfitting, some neurons are dropped while training



CNN Working Example



Model Optimizer: Adam (A variation of gradient descent)

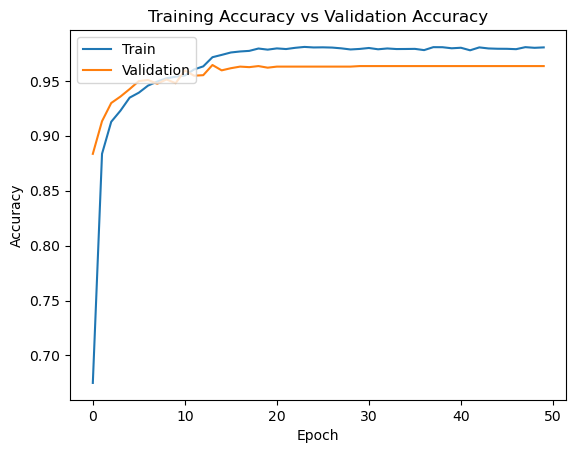
Model Loss Function: Categorical CrossEntropy (Used for multiclass classification). Here 36 classes (10 numbers + 26 alphabets) are present

Activation Functions: Softmax is used for final layer as it gives probability for each class. For all other layers, ReLU activation is used.

1. **Performance Metrics**
2. **Character Recognition model performance:**

* Training accuracy: **98.04%**
* Training loss: **0.0393**
* Testing accuracy: **98.80%**
* Testing loss: **0.0341**

Here, testing accuracy is actually more than training accuracy. This shows that model is trained well and can accurately classify alphabets and numbers.



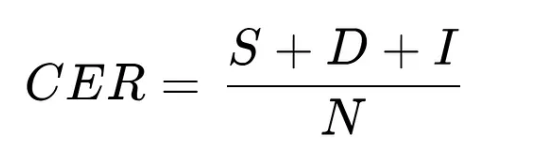
Model Training for 50 Epochs

1. **OCR Performance:**

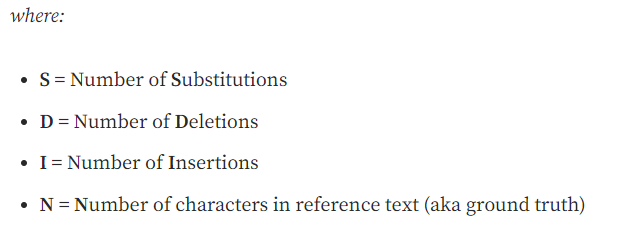
Different types of errors that can occur in optical character recognition:

****

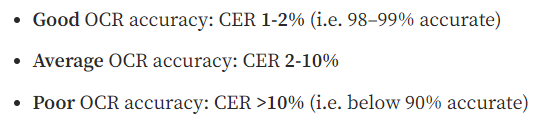
The character error rate (CER) is defined as:

****

Where,



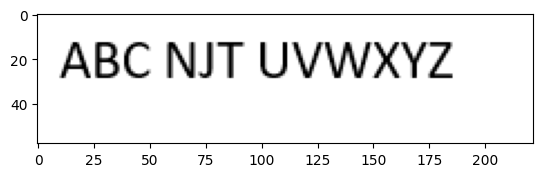
Different OCR accuracy and loss (CER) are as follows:



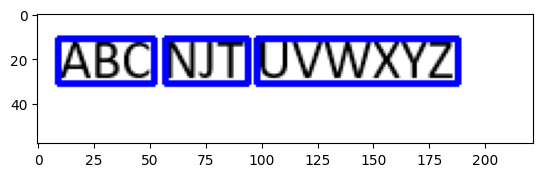
We tested for 1000 wikipedia images and found the CER to be **1.41%.** Thus, our OCR has good accuracy.

**OCR example:**

Input Image:



Word Segmentation:



Final result after character segmentation and CNN classification:



Here, OCR Identifies most of the letters correctly

**References:**

1. Sahana K Adyanthaya, 2020, Text Recognition from Images: A Study, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) NCCDS – 2020 (Volume 8 – Issue 13),
2. Ye, Q., Huang, Q., Gao, W., & Zhao, D. (2005). Fast and robust text detection in images and video frames. *Image and vision computing*, *23*(6), 565-576.
3. Hossain, M. A., & Afrin, S. (2019). Optical character recognition based on template matching. *Global Journal of Computer Science and Technology*.
4. Islam, N., Islam, Z., & Noor, N. (2017). A survey on optical character recognition system. *arXiv preprint arXiv:1710.05703*.
5. Karthikeyan, U & Muthuraman, Vanitha. (2019). A Study on Text Recognition using Image Processing with Datamining Techniques. 10.13140/RG.2.2.30668.67208.
6. Jung, K., Kim, K. I., & Jain, A. K. (2004). Text information extraction in images and video: a survey. *Pattern recognition*, *37*(5), 977-997.
7. Liu, C., Wang, C., & Dai, R. (2005, August). Text detection in images based on unsupervised classification of edge-based features. In *Eighth International Conference on Document Analysis and Recognition (ICDAR'05)* (pp. 610-614). IEEE.
8. He, T., Huang, W., Qiao, Y., & Yao, J. (2016). Text-attentional convolutional neural network for scene text detection. *IEEE transactions on image processing*, *25*(6), 2529-2541.