

---

# CS771: Introduction to Machine Learning

## Assignment 3

---

Avadhi Jindal, Kartik Jhanwar, Khyathi Vagolu, Nikhil Mehta, Sarthak Gothalyan

### Problem 3.1

Describe the method you used to find out what characters are present in the image. Give all details such as algorithm used including hyperparameter search procedures, validation procedures. You have to give a detailed explanation even if you used an algorithm/implementation from the internet – make sure to give proper credit to the person/source from where you took code/algorithm. There is no penalty for using someone else's code/algorithm but there would be heavy penalty for doing so without giving proper credit to that person/source.

### Solution

We used a Convolutional Neural Network (CNN) model for this assignment. Convolutional Neural Network is a deep learning neural network. It is specifically used for object recognition and classification. CNN uses principles of linear algebra like matrix multiplication to identify patterns within an image. A CNN consists of three layers: a convolutional layer, a pooling layer and a fully connected (or densely connected) layer. In convolutional layer, the image is converted into numerical values, which allows the CNN to interpret the image and extract relevant patterns from it. This involves a kernel checks if any feature is present in the image by performing multiple iterations. A dot product between input pixels and the filter(kernel) is calculated after each iteration. The final output from the series of dots is known as a feature map. Next comes the pooling layer. In this layer also the kernel sweeps over the input image but with reduced number of parameters. This layer reduces complexity but induces some information loss. The fully connected layer is the last layer. Based on the features extracted in the previous layers, image classification is done in this layer. All the inputs or nodes from one layer are connected to every activation unit or node of the next layer. Image classification happens in this layer.

For pre-processing the data, we did the following steps:

- Firstly, since we did not need the input images in the color scale, we converted them into grayscale images.
- Then, we extracted the edges of the images
- Lastly, we performed noise removal by erosion.

Before training, the 3 characters in the input images are cropped and put into different images.

After training, the same procedure is repeated on the test set where the cropped images yield 3 characters which are combined to form a string.

### Hyperparameter tuning:

For hyperparameter tuning, we manually changed the values and checked for their performance and accuracy. We tested models with different hyperparameters. Each model was trained on the dataset with a validation split of 0.2 for five epochs. We tried the ReLu and leaky-ReLu activation functions and found that ReLu gave higher prediction accuracy.

We found that the model with two Convolutional layers with 64, 32 filters and 4, 3 kernel sizes respectively gave the best performance.

To improve the accuracy, we performed **data augmentation** by rotating the images by 0 to 40

degrees in both clockwise and anticlockwise directions. This gave better results as compared to when the model was trained without data augmentation.

The details of our model are as follows:

**Model: "sequential"**

Layer (type)	Output Shape	Param Number
conv2d (Conv2D)	(None, 150, 116, 128)	3328
max_pooling2d (Max Pooling 2D)	(None, 75, 58, 128)	0
conv2d_1 (Conv2D)	(None, 75, 58, 64)	131136
max_pooling2d_1 (Max Pooling 2D)	(None, 38, 29, 64)	0
flatten (Flatten)	(None, 70528)	0
dense (Dense)	(None, 24)	1692696

Total params: 1,827,160

Trainable params: 1,827,160

Non-trainable params: 0

#### References:

1. <https://www.techtarget.com/searchenterpriseai/definition/convolutional-neural-network>