Low Level Design (LLD)

Automatic Number Plate Detection and Recognition (ANPR)

Written by	Sandeep Kapoor
Document version	0.1
Last revised date	22 Feb 2023

Version	Date	Autor	Comments
0.1	17 Jan 2023	Sandeep Kapoor	Final LLD

Contents

Chapter

1. Abstract & Key Words

1.1 Abstract

Computer vision is a field of study focused on the problem of helping computers to see. At an abstract level, the goal of computer vision problems is to use the observed image data to infer something about the world.

In huge systems of vehicle control like traffic management on highways or traffic management in big events and at highway tolls gathering information of the vehicles which were involved, by their number plates manually is very difficult so, AI can solve this problem using advanced computer vision algorithms.

Automatic Number Plate Detection and recognition is a technology that uses optical character recognition on images to read vehicle registration plates / number plates.

1.2 Key Words

Number Plate Detection, ANPR,

Object Detection, Object Recognition, Deep Learning
Computer Vision

Chapter

2. Introduction

Number Plate Detection is a computer vision task that involves identifying and locating vehicle registration plates from images or video streams.

This technology is used in many applications, from automated toll collection systems to surveillance and security systems. It is also used to identify stolen cars, monitor traffic violations, and detect vehicles involved in crimes.

This technology is set to revolutionize the way we interact with vehicles and ensure that roads are safer. This technology is becoming increasingly important as the number of vehicles on the road continues to increase.

Chapter

3. Basic Terminology

Image Resizing- Resizing allows you to make your image smaller or larger without cutting anything out. Resizing alters the image's dimensions, which typically affects the file size and image quality. The most common reason for resizing photos is to reduce the size of large files to make them easier to email or share online.

Convolutional Neural Network- Within Deep Learning, a Convolutional Neural Network or CNN is a type of artificial neural network, which is widely used for image/object recognition and classification. Deep Learning thus recognizes objects in an image by using a CNN.

Image Dimensions- Image dimension refers to the width and height of an image, measured in pixels.

YOLO- You Only Look Once

Chapter

4. Data Description

Here the dataset is containing two directories. Out of which one is containing images, and another is containing annotations of the respective images.

Here the word Annotations refers to the details of the bounding box of the object that is present in the image like: class, bounding box coordinates i.e., x minimum, x maximum, y minimum, y maximum.

The images in the data can be in different shapes. So, they are reshaped in the dimensions (640 * 640 * 3). And the Annotations are also scaled respectively.

Images are in different file formats like: .jpg, .jpeg, .png. And Annotations are stored in Extensible Markup Language (.xml).

The dataset is collected from [Kaggle](). As well as some numbers of images are also collected manually. And also, the respective Annotations including the bounding box, in order to get some real-life images which basically contains noise.

The main goal to perform this step is to add some more noise in data and create more generalized and robust model.

Chapter

5. Methodology

5.1 Data Collection

Data collection is always considered as the first step of every typical machine learning or Deep Learning project. Quality and size of data is one of the most important key points in every model performance.

As mentioned in the Data Description, the data is collected from Kaggle. In addition, some of the dataset is also collected manually from various sources and Annotated manually.

5.2 Data Pre-processing

Basically, collected data is in the row format. Not pre-processed at all. Providing this raw data directly to the model is always not advisable.

It may fluctuate of decrease the quality of model performance.

5.2.1 Resizing

Image resizing is a process in image processing that involves scaling an image to a desired size. This can be done by either increasing or decreasing the size of the image. It is typically used to reduce the file size of an image or to fit an image into a certain frame size.

For model training, we have to give same size images to the model. So as per the individual image size we would have to either upscale it or downscale it.

Here the chosen image size is (640* 640).

5.2.2 Rescale

Image rescaling is a process of changing the size of an image. It is typically used to resize images to fit a certain size or resolution, or to reduce the physical size of an image file. It is also used to reduce the number of colors in an image, or to change the aspect ratio of an image.

Here Image Resizing is done by scaling image pixel values from 0 to 1. It is similar to MinMaxScaler in sklearn.

It is achieved by dividing all the pixel values by 255.

5.3 Model Building

Here two type of CNN models are used.

(1) Custom CNN model from scratch
(2) YoloV5

5.3.1 Custom CNN model from scratch:

Convolutional Neural Networks (CNNs) are a type of neural network architecture that is typically used for solving computer vision tasks. The architecture of a CNN consists of an input layer, which is the raw pixel data of an image, followed by several convolutional layers, which perform convolution operations on the input image.

➔ Convolution Operation:

Convolution is a key operation in convolutional neural networks (CNNs). It is a mathematical operation that takes two inputs, one is a matrix of image pixels and the other is a matrix of weights or kernels, and produces a new matrix of outputs. In CNNs, convolutional layers apply a convolution operation to the input, passing the result through an activation function and then pooling the result into the next layer. This process allows the neural network to learn features from the input images and extract features from them. Convolutional layers are essential for image classification, as they enable the network to identify patterns in the input images.

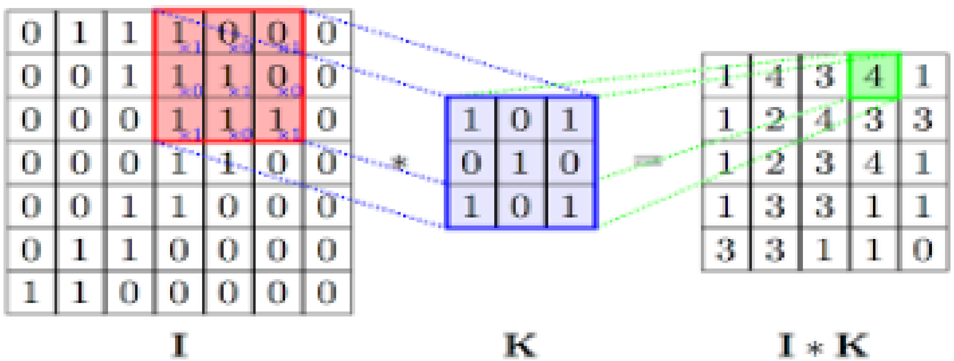


Figure 2. Represents the convolution using the filter K over the image.

➔ Max Pooling:

Max pooling is a technique used in convolutional neural networks (CNNs) for reducing the size of an input, or for extracting features from an input. It works by selecting the maximum value from a region of the input, typically a 2x2 or 3x3 region. This reduces the dimensionality of the input, allowing the neural network to better focus on the most important features in the input. Max pooling also helps to prevent overfitting, as it reduces the number of parameters that the network needs to learn.

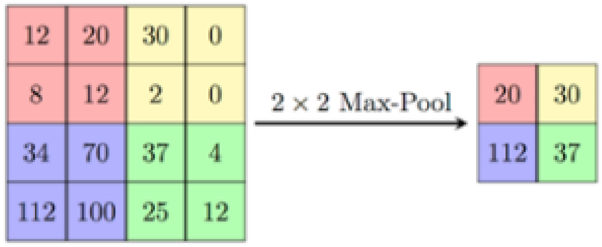


Figure 3. describes the max-pooling from the 2*2 window and reducing the dimension from 4*4 to 2*2.

➔ Batch normalization Layer:

Batch normalization is a technique used to normalize the inputs of each layer in a neural network to have a mean of 0 and unit variance, which helps the network to learn faster and generalize better.

➔ Flatten layer:

Flattening a layer in one line involves combining all the layers of a model into a single layer, usually the output layer, so that the output of the model is a single, flat vector.

➔ Dropout Layer:

Dropout is a regularization technique used in neural networks, where a proportion of nodes are randomly excluded from the network during training to reduce overfitting.

5.3.2 **YoloV5**

YOLO (You Only Look Once) is a state-of-the-art, real-time object detection system. Yolo has the capability to dominant all the others algorithms.

It is a convolutional neural network (CNN) based model that is able to detect objects in an image or video by using a single neural network. YOLO divides a given image into a grid of 13×13 cells and each of these cells is responsible for predicting 5 bounding boxes. Each bounding box consists of 5 attributes: (x, y) coordinates of the center of the box, the width and height of the box, and the confidence score of having an object in that box. In addition, the model also predicts the class probability for each box. YOLO is fast and accurate compared to other object detection models.

5.4 Model Training

In model building step only the model architecture is built.

Now we have to train the data to the model.

Before model training we would perform data augmentation.

- ➔ Data Augmentation

Data augmentation is a process used to increase the amount and diversity of data available for training in machine learning and deep learning tasks. It is a form of regularization that allows machine learning models to generalize better to unseen data. Data augmentation is typically used when the amount of training data is limited, or when the data is imbalanced. It can also be used to introduce additional variability into the training dataset and help prevent over-fitting. Data augmentation techniques include adding noise to the data, randomly flipping or rotating images, randomly cropping images, and using data augmentation algorithms such as GANs (Generative Adversarial Networks).

5.5 Database

Here the Prediction database is maintained, Prediction database refers to the database where the predicted numberplate data is stored.

Here SQLite is chosen as the database. It is a lightweight database we can use along with python.

From the API we also can retrieve / download the data in the form of CSV file. And also we can Reset session. Where Reset Session refers to delete all the records from the database table.

Chapter

6. Result and Discussion

6.1 Loss Metrics

Object Detection is basically a Regression problem so here the chosen loss metrics is Mean Squared Error.

Since Regression output is the four edges of the object, the error is mean squared distance between original and predicted bounding box edges, x min, x max, y min, y max respectively.

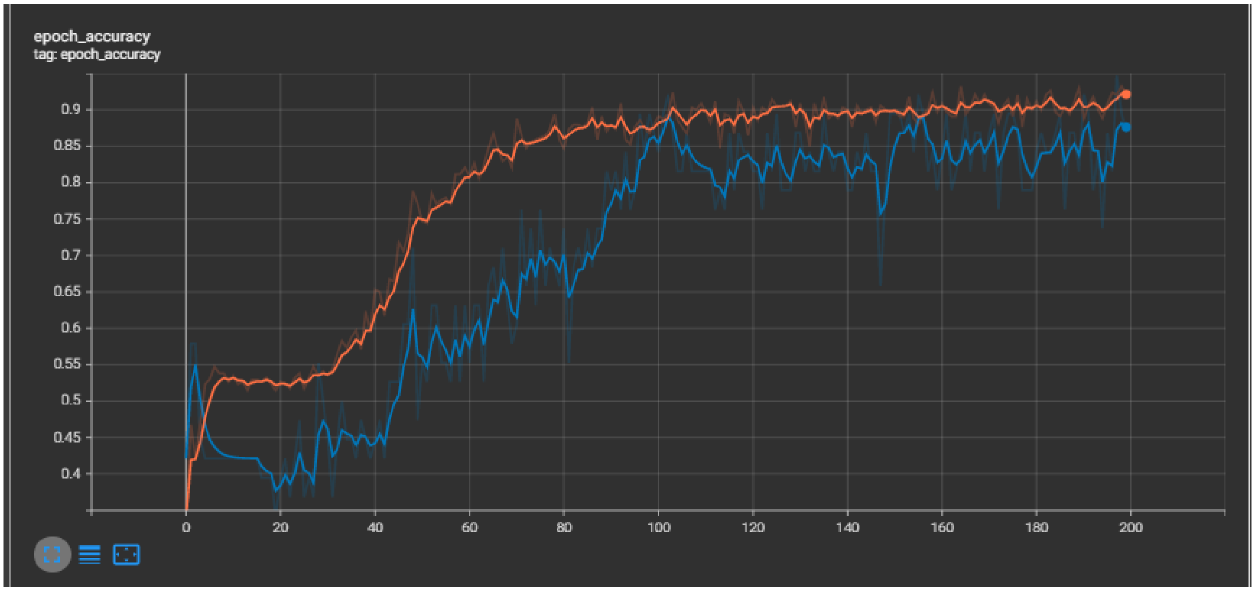
$$MSE = \frac{\sum(y_i - \hat{y}_i)^2}{n}$$

Where:

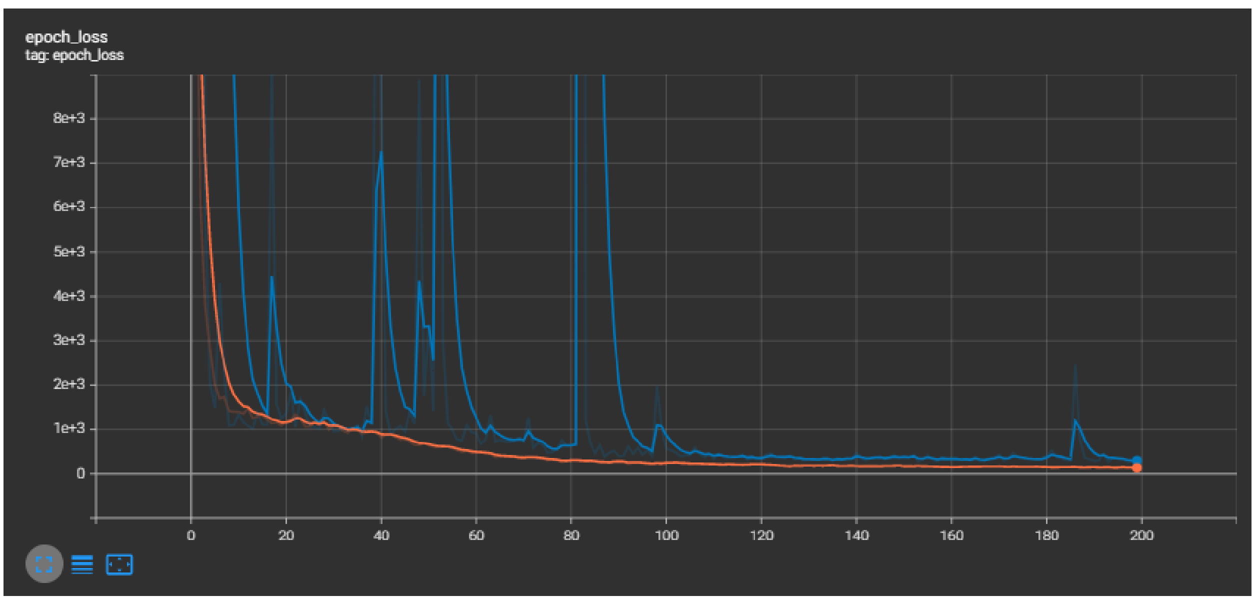
- y_i is the i^{th} observed value.
- $\hat{y}_i$ is the corresponding predicted value.
- n = the number of observations.

6.2 Accuracy & Loss Evaluation

Training Accuracy



epoch_accuracy
tag: epoch_accuracy
0.9
0.85
0.8
0.75
0.7
0.65
0.6
0.55
0.5
0.45
0.4
0
20
40
60
80
100
120
140
160
180
200



epoch_loss
tag: epoch_loss
8e+3
7e+3
6e+3
5e+3
4e+3
3e+3
2e+3
1e+3
0
0
20
40
60
80
100
120
140
160
180
200