REPORT

ML BASED NUMBER PLATE RECOGNITION MODEL USING COMPUTER VISION

By -MAYANK YADAV 22112064
MD FARHAN 22112065
KOUSIK KUMAR BARNAWAL 22112056

1. Introduction

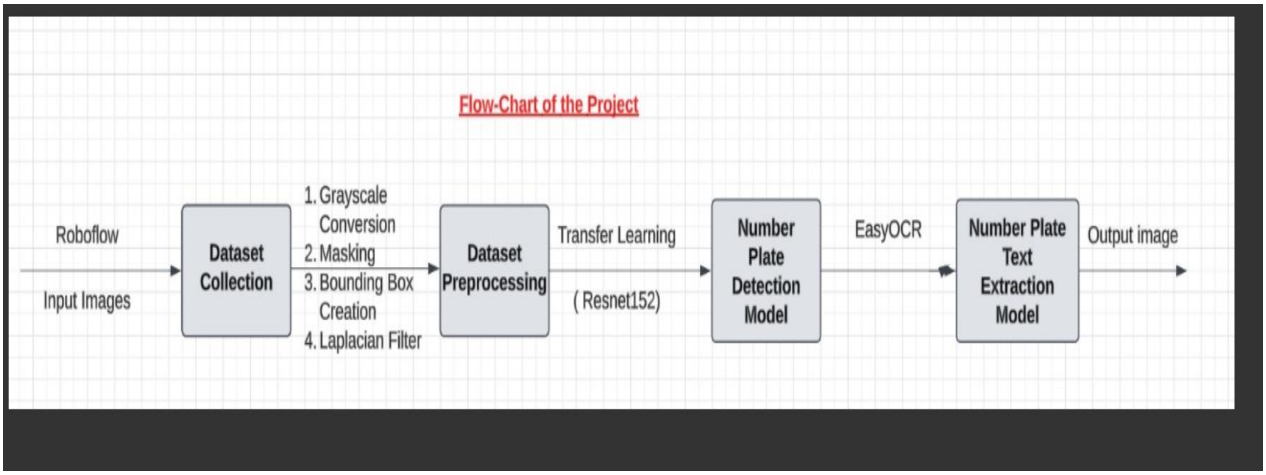
- Objective: This project aims to develop a computer vision model capable of detecting bounding boxes around specific regions in images and extracting any text within these detected regions. The purpose is to create a streamlined approach for localizing and extracting text, which could be used in various fields such as automatic number plate recognition or document analysis.
- Dataset Overview: The dataset consists of labeled images, each with corresponding bounding box coordinates stored in text files. These labels provide object location and size information, which the model uses to learn to localize and identify regions containing text.
- Project Scope: The scope includes data preprocessing, bounding box localization, text detection, and text extraction using optical character recognition (OCR). The model will be trained on a subset of the data to improve text detection accuracy.

2. Libraries Used

- NumPy: For numerical operations, primarily array manipulation.
- Pandas: For data management and analysis, though minimally used here.
- Warnings: To filter out unnecessary warnings during execution.
- OpenCV: For image processing, transformations, and visualization.

- Keras OCR and EasyOCR: Text detection libraries that facilitate OCR tasks.
- Pytesseract: An OCR library for extracting text from images.
- Matplotlib: For visualizing model training and image results.

3. Methodology



Data Preprocessing: A preprocessing function loads images and corresponding bounding box labels, then resizes images to a target size (224x224). Grayscale conversion and Laplacian edge detection are applied to highlight text edges. Bounding box coordinates are adjusted based on the new image dimensions, and binary masks are created for precise focus on the text region.

Bounding Box Prediction: A deep learning model based on ResNet152 extracts features from images. This model has been modified to create bounding box coordinates through a series of convolutions and transpositions, which assist in detecting areas likely to contain text.

Text Detection and Extraction: After predicting bounding boxes, the selected region of interest (ROI) is processed with various morphological operations, and text is extracted using OCR libraries, primarily EasyOCR and Pytesseract.

4. Model Architecture

Layer (type)	Output Shape	Param #
resnet152 (Functional)	(None, 7, 7, 2048)	58,370,944
conv2d_6 (Conv2D)	(None, 7, 7, 512)	9,437,696
batch_normalization_22 (BatchNormalization)	(None, 7, 7, 512)	2,048
conv2d_transpose_21 (Conv2DTranspose)	(None, 14, 14, 256)	1,179,904
batch_normalization_23 (BatchNormalization)	(None, 14, 14, 256)	1,024
conv2d_transpose_22 (Conv2DTranspose)	(None, 28, 28, 128)	295,040
batch_normalization_24 (BatchNormalization)	(None, 28, 28, 128)	512
conv2d_transpose_23 (Conv2DTranspose)	(None, 56, 56, 64)	73,792
batch_normalization_25 (BatchNormalization)	(None, 56, 56, 64)	256
conv2d_transpose_24 (Conv2DTranspose)	(None, 112, 112, 32)	18,464
batch_normalization_26 (BatchNormalization)	(None, 112, 112, 32)	128
conv2d_7 (Conv2D)	(None, 112, 112, 3)	99

ResNet152 as Base Model: The ResNet152 architecture serves as a base model for feature extraction, which allows for learning complex patterns in images. Only the last few layers of ResNet152 are trainable to balance feature extraction capabilities with training efficiency.

- Additional Layers:
- A series of Conv2D layers followed by Conv2DTranspose layers (upsampling) add decoding capability.
- Batch normalization layers are used after each convolutional operation to stabilize training.
- The final output layer, a Conv2D layer with a sigmoid activation, is designed to output a mask for text localization.

5. Model Compilation and Training

- Loss Function: The model uses Intersection-over-Union (IoU) loss, which penalizes the model when the predicted bounding box diverges from the ground truth.
- Metric: IoU metric is also used for monitoring training progress.
- Training Setup: The model is compiled with the Adam optimizer. The training process includes 80 epochs, with an 80-20 training-validation split.
- Data Input: Images are input in grayscale format, resized to (224, 224), and normalized to a range of [0, 1].

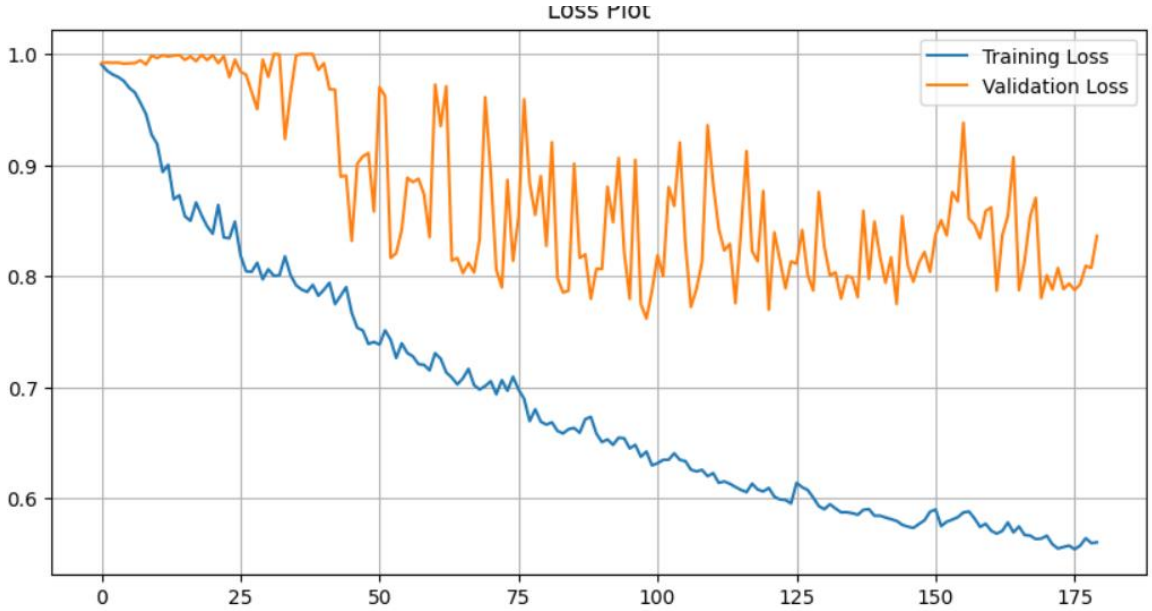
6. Testing and Evaluation

- During testing, images undergo similar preprocessing as the training images, with the trained model predicting bounding box coordinates around text regions. Predicted bounding boxes are evaluated by overlaying them on the original images, and the IoU score helps in gauging localization accuracy.
- Evaluation Metric: IoU metric is utilized to quantify the overlap between predicted and ground-truth bounding boxes, indicating localization accuracy.

7. Results and Observations

Loss and Accuracy Plot: Training and validation loss curves reveal that the model achieved a consistent decrease in loss, indicating good training performance. IoU scores also reflected satisfactory bounding box localization in validation images.

Text Detection Accuracy: OCR results for detected regions showed high accuracy in text extraction for clear and visible text. However, challenging images with blurred text or varied lighting conditions occasionally led to suboptimal extraction results.



8. Conclusion

- The project successfully created a model that localizes and extracts text in images by predicting bounding boxes around text regions and using OCR for text extraction. The model is efficient for real-time applications but could benefit from further training data or fine-tuning to handle complex backgrounds and lighting.
- Future Scope: Enhancements could include adding more layers to the model or experimenting with a higher-resolution input for better feature extraction.

9. Contributions by Team Members

- Team Member Mayank Yadav : Developed the whole idea structure of preprocessing ,modelling and helped in error debugging .
- Team Member Md Farhan : Handled the dataset loading and preprocessing of images and analysing the result of text detection by different model like easy ocr and pytesseract
- Team Member Kousik Kumar barnwal : Testing on different models such as wpodnet,resnet and extract different ideas from different articles.