ITNPAI1 Deep Learning
Spring 2023

Street Litter Detection Model

Report

Bipin Babu (3061287),
Lakshmi Narayanan Namboothiri (3082182)

Introduction

Street litter has been a growing problem that significantly contributes to environmental pollution [1]. Litter waste in the streets has increased drastically since the COVID-19 pandemic [1]. From food packaging to cosmetic litter, a wide variety of litter can be found on the streets.

This project aims to build a deep-learning model for street litter detection. If successful, it could aid city cleaning initiatives, provide insights for litter control and prevention research, and enhance monitoring.

We chose two cities to perform this task. One of the cities is **Falkirk, Scotland, UK** (**Coord: 56.0019° N, 3.7839° W**), and the other city is **Punalur, Kerala, India (Coord: 9.0176° N, 76.9261° E).** Since both of us (the contributors of this project) are from the same hometown, Punalur, Kerala, India**,** we decided to choose Falkirk, Scotland, UK as one of the cities, which is the current residence city of one of the contributors, Lakshmi.

Dataset Collection

More details on data collection are highlighted in the file:
'2023_Spring_Assignment_AI1.ipynb'.

Description of the proposed solution

Approach 1: Object Detection using a region proposal algorithm.

Collecting our own dataset for the project inspired us to adopt a region proposal-based approach for litter detection, offering better control and flexibility over image manipulation.

Pre-processing tasks

a. **Normalisation**
   Normalization was performed to ensure that all pixel values in the images were within a range of 0-255 for consistency throughout the task.

b. **Sharpening**

   Images were sharpened using a (3,3) Kernel to improve edge detection and enhance the accuracy of litter detection [2].

   Kernel Used:
   [[0 -1 0]
   [-1 5 -1]
   [0 -1 0]]

c. **Histogram Equalization**
   Histogram equalization was performed to enhance the contrast of the images.

For the region proposal operation, the 'Selective Search algorithm' [3] was employed as it is easily implementable and readily available in OpenCV. The images were resized, with the longer side being set to 600px to preserve the aspect ratio and to reduce the time consumption of processing images.

The Selective Search algorithm generated candidate regions for each image, which were then compared to ground truth bounding box values using the Intersection Over Union (IOU) value. This value signifies the area of the overlapping region between the proposed candidate region and the annotation file's ground truth bounding box values, indicating how close they are. The IOU calculation code was adopted from [4].

Based on the IOU threshold value set (0.7 for litter and 0.3 for background), each proposed region is cropped out of the pre-processed image and classified as litter or non-litter. This classified data is passed to a classifier, VGG16, for predicting the class of future proposals from unseen images.

The VGG16 output classification probability is used to draw a bounding box in the new image obtained from the selective search, indicating a potential litter when the probability is above a certain set threshold.

VGG16 Classifier

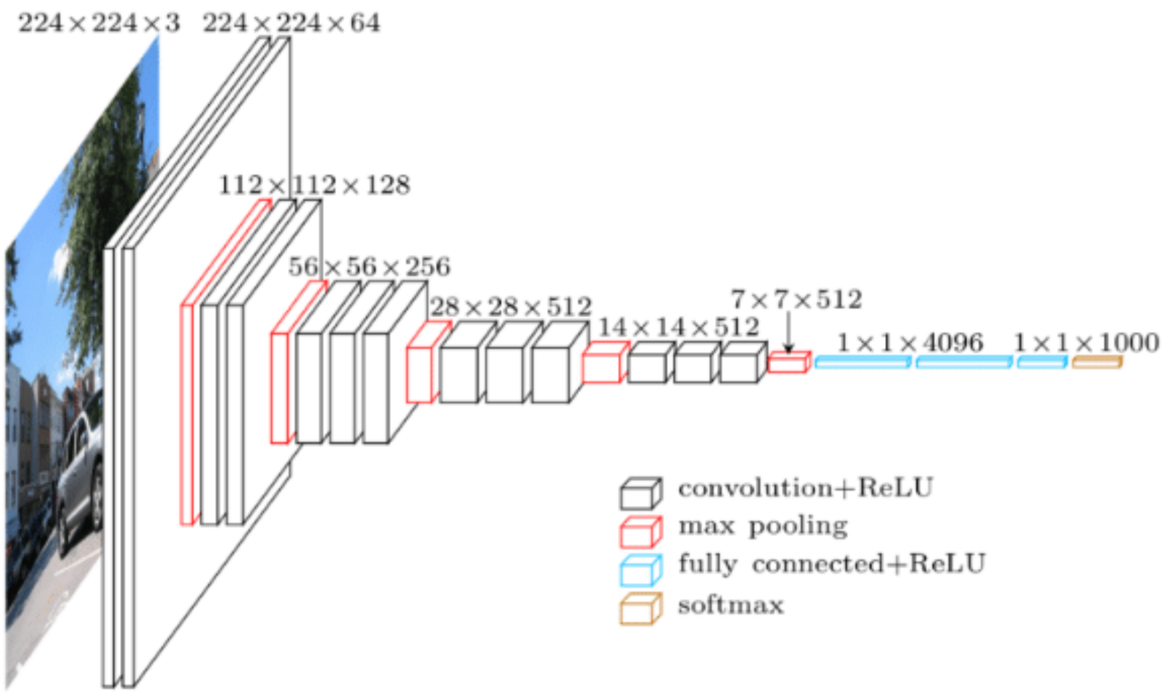

Figure 1: VGG16 architecture [5]

The VGG16 classifier is pre-trained on the ImageNet dataset with an accuracy of 92.70%, making it suitable for the task at hand.

For the street litter detection task, we froze the weights of all the layers of the VGG16, and a flatten layer, a dense layer with ReLU activation function followed by a dense layer with sigmoid activation function, was added. The flatten layer converts the 2D outputs of VGG16 to a one-dimensional vector. The Dense layer with the ReLu activation function ensures that non-linear complex properties of the data are learned at this stage, and the final Dense layer produces a value between 0 and 1, i.e., the probability of being a litter.

The initial approach was to freeze the first 15 layers of the model and utilise the second last layer of the model to add a dense layer with the sigmoid activation function. However, the model was overfitting with this approach and had to be modified with multiple trials to find the current configurations. After spending a considerable amount of time debugging and trying out various configurations, we had already moved on to Approach 2 using the YOLO family of algorithms before getting a fair accuracy with the current configurations.

Approach 2: Object Detection using the YOLO family of algorithms.

YOLOv8 is a highly accurate computer vision model that simplifies the coding process for developers [7]. It offers three versions: Nano, Small, and Medium, and uses stochastic gradient descent as the optimizer with an IOU threshold of 70%. Basic data augmentation techniques were used for these models, including mosaic and flipping. The batch size is 16, with a constant input size of 640. The learning rate is defined as 0.01.

Results

1. Training Results

a. Region-based approach on city Punalur

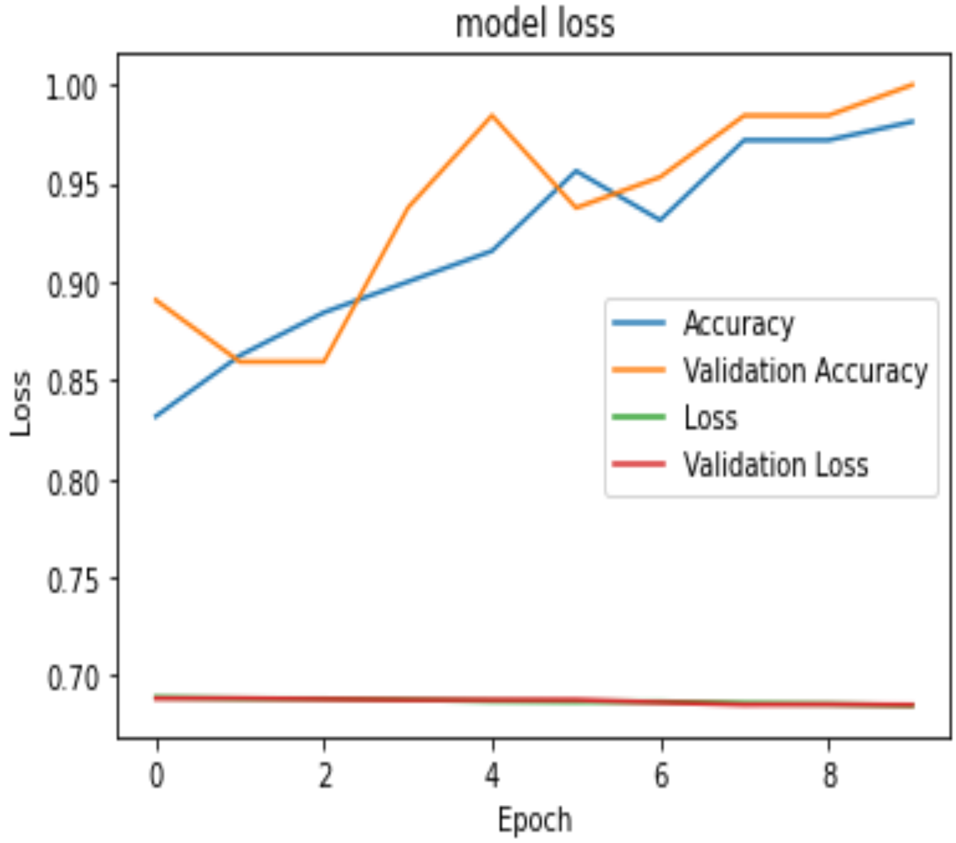

Figure 2: Region-based approach training on city Punalur dataset

b. Region-based approach of Falkirk

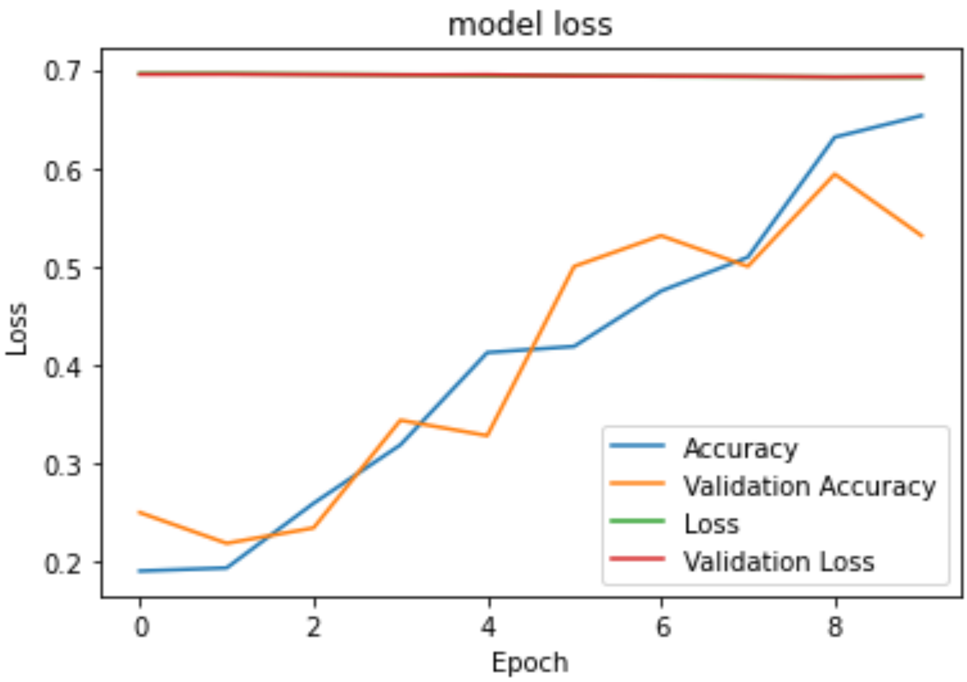

Figure 3: Region-based approach training on city Falkirk dataset

2. Testing Results

2.1 Region-based approach testing

2.1.1 Testing Falkirk Trained model on Falkirk Data

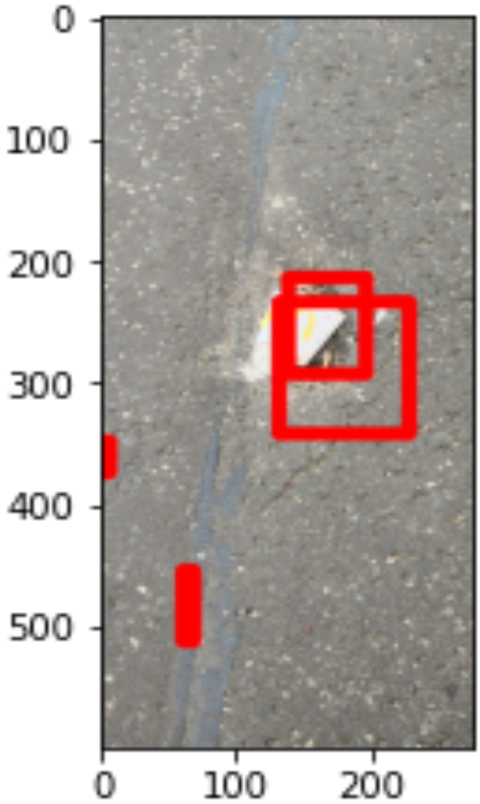

Figure 4: Falkirk model Prediction on Falkirk data

2.1.2 Testing Punalur Trained model on Punalur Data

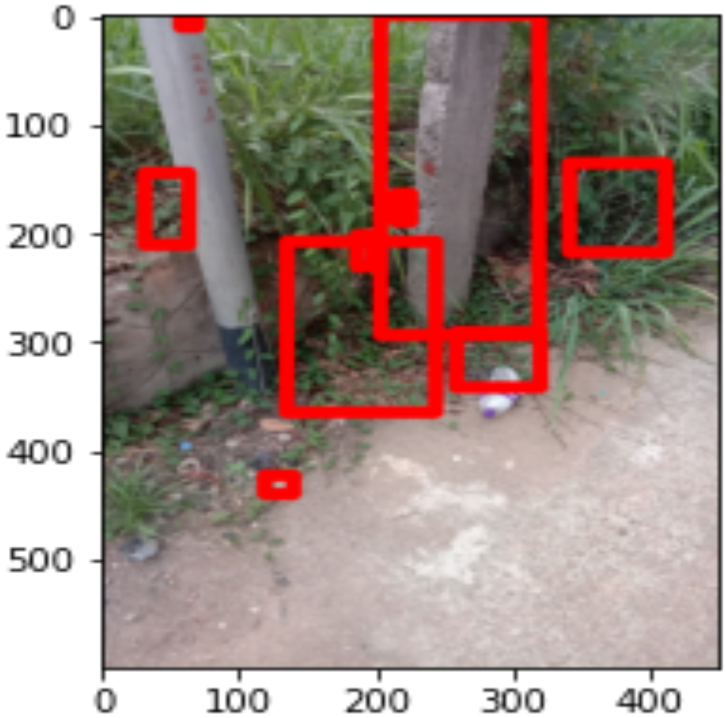

Figure 5: Region-based approach prediction of Punalur model on Punalur data

2.1.3 Testing Falkirk trained model on Punalur Data

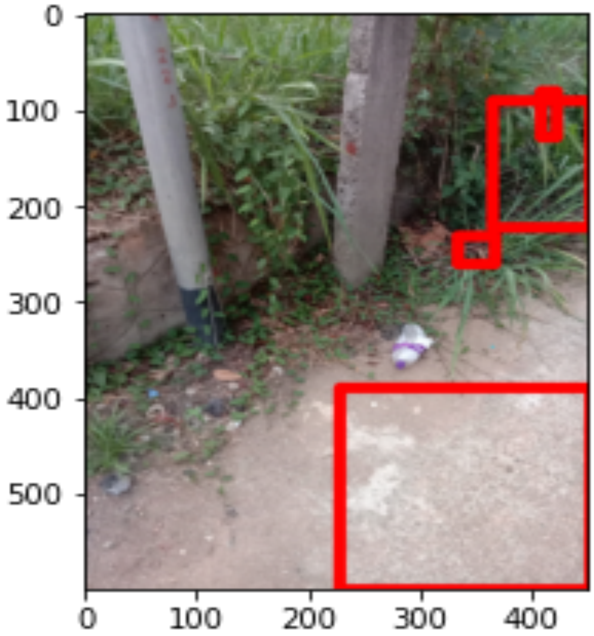

Figure 6:Falkirk trained model on Punalur data

2.1.3 Testing Punalur trained model on Falkirk Data

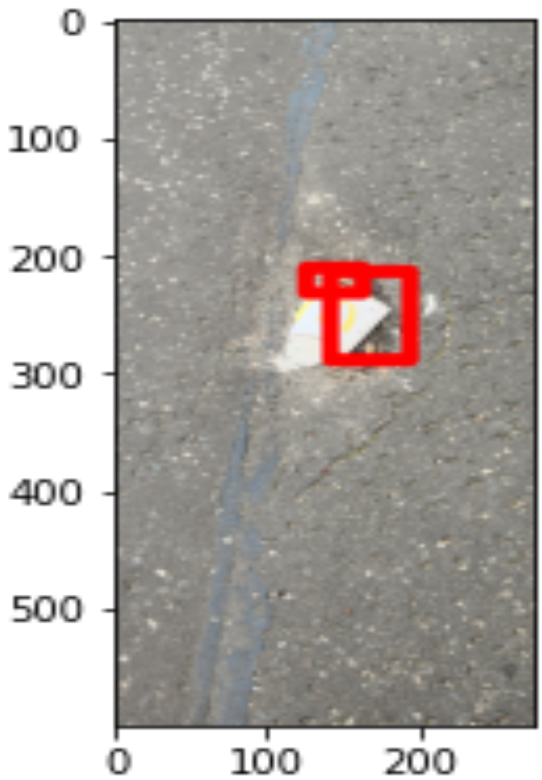

Figure 7; Punalur trained model on Falkirk data

2.2 YOLO testing

Figure 8: Falkirk trained medium model on Falkirk data

Figure 9 : Punalur small trained model on Punalur data

Figure 10: Falkirk trained medium model on Punalur data

Figure 11: Punalur trained small model on Falkirk data

Discussion on Results

RCNN Training/Testing process

- **Training RCNN on city Punalur**: It can be seen in Figure 2 that the Training accuracy and Validation accuracy have improved with an increase in epochs. However, after 10 epochs, the training accuracy has converged to 98%, and the validation accuracy of ~99%, suggesting that the model is overfitted.

- **Training RCNN on city Falkirk:** It can be seen in Figure 3 that after 10 epochs, the model gives a training accuracy of 65% and a validation accuracy of 53%.

- **Testing RCNN trained on Falkirk data with Falkirk data:** Figure 4 shows that the model can detect the litter properly but with a constraint in the number of boxes.

- **Testing RCNN trained on Punalur data with Punalur data:** Figure 5 shows that the prediction from this model is unreliable and is generalizing non-litters as litter.

- **Testing RCNN trained on Falkirk data with Punalur data:** It can be seen from Figure 6 that the model is not predicting the litter accurately on a different city data, even though it was predicting well on the same city data.

- **Testing RCNN trained on Punalur data with Falkirk data:** Even though the model was doing poorly in generalized prediction, Figure 7 shows that the model detects the litter fairly accurately.

YOLO Training/Testing process

The Nano and Small models are trained for 22 epochs on the Punalur dataset and tested on both sets of data, which has an mAP of 0.537 and 0.324, respectively. Both models performed well by updating the confidence level to 0.30 for nano and 0.25 for small, respectively.

The Medium model is trained for 30 epochs on the Punalur dataset and tested on Falkirk and has an mAP of 0.496. By tuning the confidence level to 0.15, the model was predicting well.

The Nano model is trained for 22 epochs on the Falkirk dataset and tested on both sets of data, which has mAP of 0.82. By tuning the confidence level to 0.25, the model was predicting well.

The small model is trained for 22 epochs on the Falkirk dataset and tested on both sets of data which has an mAP of 0.822. By tuning the confidence level to 0.35, the model was predicting well.

The Medium model is trained for 30 epochs on Falkirk dataset and tested on both set of data which has an mAP of 0.895. It achieves the best results in detecting objects by setting a confidence score of 0.40.

Training the models with Falkirk and Punalur data together improves accuracy, with the Nano model achieving an mAP of 0.906 and the Small model achieving an mAP of 0.822.

From this result, it is clear that models trained using Falkirk pictures showed better accuracy than Punalur. The quantity and diversity of data collected from Falkirk surpass that of Punalur. The Falkirk medium model and Punalur small model performed best in each section.

Since the result clearly states that a variety of data can enhance the accuracy of the model, we have tried training Falkirk and Punalur data together and train models which show robustness.

Conclusion

The initial approach we carried out by training a classifier based on the region proposal from selective search was not performing well on both cities' data due to overfitting and lack of enough samples. Moreover, the current configuration of this approach was obtained only towards the final phase of the project, which resulted in less tuning and experimentation to achieve a good metric and prediction.

On the other hand, the YOLO family of algorithms which were employed for the task are good at generalising litter and performs well on both cities' data irrespective of where it is tested on. This shows the robustness of YOLO algorithms and their generalised prediction capabilities.

References

[1] A. F. Kasra Karimi, “The Issues of Roadside Litter: A Review Paper,” doi: https://doi.org/10.4236/cus.2021.94046. [Online]. Available: https://www.scirp.org/(S(czeh2tfqw2orz553k1w0r45))/journal/paperinformation.aspx?paperid=113864

[2] A. Chris, *Machine Learning with Python Cookbook*. O’REILLY, pp. 131–132.

[3] “OpenCV Selective Segmentation.” [Online]. Available: https://docs.opencv.org/3.4/d6/d6d/classcv_1_1ximgproc_1_1segmentation_1_1SelectiveSearchSegmentation.html

[4] ITNPAI1 - Deep Learning for Vision and NLP, Practical 9: R-CNN

[5] GG16 architecture: Leaf App: Leaf recognition with deep convolutional neural networks - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/VGG16-architecture-16_fig2_321829624

[6] "Everything to know about VGG16." [Online]. Available: https://medium.com/@mygreatlearning/everything-you-need-to-know-about-vgg16-7315defb5918

[7] https://learnopencv.com/ultralytics-yolov8/

Word Count [1278 Excluding Headings, Subheadings, Captions and References]