Topics in Artificial Intelligence
Convolution Neural Network
CS-256 - Section 2

Prof. Mashhour Solh

Milestone 3

Group B

Joel Alvares	-	013714415
Rakesh Nagaraju	-	014279304
Charulata Lodha	-	014521182
Vidish Naik	-	014515358

Fall 2019

Final Project Report

1. Abstract

This project aims at detecting Pedestrians by implementing the concept of Repulsion Loss[1], based on faster RCNN and is aimed to recure the thesis "Repulsion loss" CVPR 2018. In Real-world scenarios, detecting Pedestrians can be a challenging task especially in a crowd, as many Individual pedestrians can gather around and occlude each other. As stated in the thesis "Repulsion loss" CVPR 2018, 'Repulsion Loss is a novel bounding box regression loss specifically designed for crowd scenes'. Two key points drive this loss: the attraction by target and the repulsion by other surrounding objects. The repulsion term prevents the proposal from shifting to surrounding objects thus handling the Occluded pedestrian's scenario. In this paper, we first explore various state-of-the-art papers on Pedestrian Detection. Then, we implement Repulsion Loss based on Faster RCNN and via experiment detect pedestrians in a crowded scene. We also compare the Mean Average Precision, PR curves for both Repulsion Loss implemented on SSD (Single Shot Detector) and Repulsion Loss on Faster RCNN.

2. Introduction

Object detection, a part of Computer Vision (CV) is used to classify objects like animals, humans, cars, trees and many more in an image or video. Pedestrian detection, which is a subfield of object detection, identifies humans in an outdoor environment. Based on feature extraction, it is possible to detect the presence of humans in an image or a video.

The core challenge of Pedestrian detection lies in accurately detecting pedestrians in complex backgrounds while handling real-world scenarios such as crowd occlusion. Many times detectors that work well on detecting humans suffer inaccuracy, due to occlusion. The detected bounding box encompasses multiple humans rather than generating individual bounding boxes for each human. In pedestrian detection, crowd occlusion constitutes the majority of occlusion cases. The reason is that in application scenarios of pedestrian detection, e.g., video surveillance and autonomous driving, pedestrians often gather together and occlude each other.

In the traditional approach, a sliding window extracts features of the image and feeds it to a classifier. The contents of the sliding window are passed to a classifier where it extracts the features in the window and then classifies it accordingly. The main drawback of a sliding window is that it requires multiple passes over a single image with varying window sizes.

With new high-level classifiers such as Convolution Neural Networks(CNNs), the classification has to be done for every window. This results in the calculation of more than a million weights for each window which is computationally expensive and results in slower classification. Also, as a result of sliding windows over the image, there are multiple bounding boxes of a single object which are not well defined and will cause a high background and foreground imbalance.

Through the literature review, we analyze some state-of-the-art papers on Object Detection. Then, we implement the concept of Repulsion Loss[1] based on Faster RCNN.

Our main contributions are as follows:

- First, we executed the "Repulsion Loss: Detecting Pedestrians in a Crowd" (exactly as given in the paper [1]), with Single Shot Detector(SSD) on Google Colab for 10k images of the VOC2007 dataset using an Nvidia Tesla K80 GPU, PyTorch 0.3.1 and TorchVision 0.2.0. to understand the overall flow as shown in the flowchart in **Figure 5**.
- We then executed the "Repulsion Loss: Detecting Pedestrians in a Crowd" for 10k images of the VOC 2007 dataset on p2.xlarge instance of AWS having Nvidia Tesla K80 GPU, PyTorch 0.4.0 and TorchVision 0.2.0 with overall flow as shown in the flowchart in **Figure 6**.
- With the proposed repulsion losses, a crowd-robust pedestrian detector using Faster RCNN is trained end-to-end, with the intention of increasing the accuracy to an already existing Single Shot Detector on Pascal VOC 2007 Dataset.
- We also generate PR curves, ROC curves to analyze the results. It should also be noted that the detector with repulsion loss significantly improves the detection accuracy for occlusion cases, highlighting the effectiveness of repulsion loss.

This detector has a wide range of applications ranging from the Intelligent Video Surveillance System, Collision Avoidance System (CAS) in autonomous self-driving vehicles, etc. as it provides the fundamental information for semantic understanding of the video footage.

3. Literature survey

3.1 Repulsion Loss: Detecting Pedestrians in a Crowd [1]:

The paper starts with analyzing the reasons behind the failure of previous CNN based approaches. Occlusion is the reason for 60% missed detections whereas in case of false positives indicating it is the main factor affecting the performance of the baseline detectors. In the case of false positives, occlusion contributes to about 20% of the cases. The preliminary analysis concludes that the performance of pedestrian detectors drop due to crowd occlusion. To overcome the problem of occlusion, the paper uses a ResNet-50 network as the backbone which is a lighter and faster network as compared to VGG-16. ResNet-50 is not used in detecting pedestrians as it has difficulty in detecting and localizing small pedestrians because of its high downsampling rate. The paper gets around this by using a dilated convolution and reduced the final feature map to 1/8th of the input size. It also proposes a loss function: L = L Attr + α ∗ LRepGT + β ∗ LRep Box.

The paper uses IoU or IoG over SmoothL1 for the repulsion terms is because both IoU and IoG are bound by the range [0, 1] whereas SmoothL1 is not bounded by a similar range and it is boundless. SmoothL1 will cause the predicted box to be as far away as possible from neighboring ground-truths whereas the paper aims to minimize the overlap of the predicted box with other ground-truths. Furthermore, IoG is adopted in this approach because the bounding box regressor might enlarge the bounding box size to increase the denominator in IoU thereby reducing the loss. Choosing IoG will keep the denominator constant. The paper achieves a 79.8 mean Average Prediction (mAP) which is about 3.5 mAP more than the baseline.

3.2 Part-Level Convolutional Neural Networks for Pedestrian Detection Using Saliency and Boundary Box Alignment [2]:

This paper deals with minimizing the loss of body parts due to the proposal shift problem by implementing part level CNN approach having two subnetworks: Detection & Alignment. For removing false positives it uses saliency in Detection sub-network (DSN) and then combines FCN & CAM for deep feature extraction and successfully recalls the loss of body parts using part level CNN. The experiments are based on the Caltech USA dataset, INRIA dataset, and ETH dataset. Also, considering efficiency, only 3 body parts are considered: head, torso & legs instead of 5. This new approach has shown a 10% accuracy improvement in terms of log average miss rate at false position per image (FPPI) over their previous work where only FCN was utilized. By solving the proposal shift problem, the experiment confirmed that FPPI decreases 12.40% when the bounding box alignment is applied.

PART A: Detection Framework
To obtain detection proposals, fast pedestrian detection is performed based on region proposal network(RPN). Convolutional units, one fully-connected (FC) layer, and one global max-pooling (GMP) layer is used for classification and localization.
For detection proposals, the combined loss function is optimised: L = Ld + Ls,
where Ld and LS are losses of the detection network and of the saliency network, respectively.
Secondly, the effective removal of false positives is done using saliency by assigning different weights to pedestrians & backgrounds. It updates the class probability (score) using the saliency map as follows:

$$f_w(b) = f(b) * w_f \qquad w_f = \begin{cases} 1 & if \quad f(b) > th_b \\ \frac{1}{N}\sum_{x,y\in b} s(x,y) & otherwise, \end{cases}$$

where fw(b) is a class score, b is bounding boxes of proposals, s(x,y) is saliency scores in the position (x,y), and f(b) is class scores of the selected bounding box. thb is the threshold value for distinguishing between foreground and background.

PART B: Alignment Framework
In alignment with sub-network, it then successfully recalls the lost body parts by boundary box alignment using a part-level detector. Out of 4 root level detectors, one detects the root position of pedestrians and three part-level detectors are for human body parts: head, torso, and legs. By employing a localization feature of CNN: FCN & CAM, it generates confidence maps to infer accurate pedestrian location.

3.3 Object as Points [3]:

This paper is objected at using points as a key-point estimation to detect Objects. Successful object detectors achieve detection, by identifying objects as axis-aligned boxes in an image. This process is a nearly exhaustive list of potential object locations and classify each, which seems wasteful, inefficient, and requires additional post-processing. This paper takes a different approach. It models an object as a single point (the center point of the bounding box). This approach named, CenterNet, uses key-point estimation to find center points and then regresses to other object properties, such as size, 3D location, orientation, and even pose estimation. It is end-to-end differentiable, simpler, faster, and more accurate than corresponding bounding box based detectors.

The input image is taken as, I ∈ RW×H×3, where width W and height H. Next, a key-point heat map is produced, Yˆ ∈ [0, 1] W R × H R ×C, where R is the output stride and C is the number of key-point types. A prediction Yˆ x,y,c = 1 corresponds to a detected key-point, while Yˆ x,y,c = 0 is considered to be background. Here, several different fully convolutional encoder-decoder networks are used to predict Yˆ from an image I: A stacked hourglass network, up convolutional residual networks (ResNet), and deep layer aggregation. CenterNet then builds on successful key-point estimation networks, finds object centers, and regresses to their size, as seen in the below **Figure 3**.

At inference time, the peaks in the heatmap for each category are extracted independently. Then the detector detects all responses whose value is greater or equal to its 8-connected neighbors and keeps the top100 peaks. Next, key-point values are used as a measure of its detection confidence and the bound box is estimated. Equipped with state-of-the-art keypoint estimation network, Hourglass-104 and multi-scale testing, the network achieves 45.1% COCO AP at 1.4 FPS. Image representation of Objects as Points in **Figure 3**.

3.4 Center and Scale Prediction (CSP): A Box-free Approach for Object Detection [4]:

As the name suggests, in this paper we will go through an approach that differs from the conventional approaches in which object detection is carried out with the help of sliding window classifiers or through anchor box-based predictions. Center and Scale Prediction is a box-free object detection approach which is divided into two parts: Detection of Center and Determination of Scale. Before the center and scale of the objects are determined, a feature extraction module that consists of 3 stages is used, to generate a feature map. The fused result of these multi-scale feature maps is fed to a detection head which is nothing but a 3x3 convolutional layer followed by two 1x1 convolutional layers to predict the center and scale of the object. Additionally, an offset prediction branch can be in parallel with the two 1x1 convolutional layers to slightly adjust the center location.
The loss function is a sum of the loss functions used to determine the center, scale, and offset.

$$L = \lambda_c L_{center} + \lambda_s L_{scale} + \lambda_o L_{offset}$$

Here, λc, λs, λo are the weights with respect to each of the loss functions.

Classification Loss (Lcenter): The following loss function is used to predict the center of the object where p is the probability indicating the objects center while y specifies the correct label, K is the number of objects in the image and α, γ are the hyperparameters.

$$L_{center} = -\frac{1}{K}\sum_{i=1}^{W/r}\sum_{j=1}^{H/r} \alpha_{ij}(1-\hat{p}_{ij})^{\gamma} log(\hat{p}_{ij}),$$

where, $\hat{p}_{ij} = \begin{cases} p_{ij} & \text{if } y_{ij} = 1 \\ 1 - p_{ij} & \text{otherwise,} \end{cases}$ & $\alpha_{ij} = \begin{cases} 1 & \text{if } y_{ij} = 1 \\ (1 - M_{ij})^{\beta} & \text{otherwise.} \end{cases}$

Scale Prediction (Lscale): It uses SmoothL1loss function in order to estimate the scale of the object.
Offset Prediction (Loffset): Similarly, it uses the SmoothL1 loss function to determine the offset by which the centers need to be adjusted.

3.5 Comparison of the approaches discussed above, in a Box.

Approaches	Methodology	Scale Variation	Occlusion	Accuracy
Repulsion-Loss: Detecting Pedestrians in a Crowd[1]**.**	Modified Single Shot Detector (SSD + RepLoss)	Handled by dilated convolution and reducing feature map to 1/8th size of input.	Occluded pedestrians are detected separately	Achieves MR-2 of 4.0 using RepLoss + CityPersons dataset
Center and Scale Prediction[2]**.**	Uses a CNN to generate a feature map. This is then fed to individual networks to predict the center and scale	Objects of various scales will be detected based on the dataset used while training.	Occluded pedestrians with full overlap will have a different scale and in case of partial overlap, even the center points will differ.	Achieves MR-2 of 4.5% with Caltech dataset and in City Persons an MR-2 of 3.8%.
Object as Points[3]**.**	A stacked hourglass network, up convolutional residual networks (ResNet), and deep layer aggregation is used for key estimation and then detector finds object centers and regresses to their size	Objects of various scales will be detected based on the dataset used while training.	Occluded objects will be considered as a single object.	45.1% COCO AP at 1.4 FPS
Part level CNN for Pedestrian Detection Using Saliency and Boundary Box Alignment[4]**.**	Saliency is used to remove false positives. FCN and CAM are combined to extract deep features.	Proposal-and-classification approach to detect pedestrians with multi-scales.	Partially-occluded or low-resolution pedestrian detection is possible using a part-level detector.	The proposed method achieves a comparable performance of 10.34% to state-of-the-art in a partially-occlusion INRIA dataset.

4. Methodology

Through the literature review, we analyzed the most dominant aspects for improvement of Pedestrian Detection namely Centre-Point Estimation, Bounding Box Alignment, Center and Scale Prediction, and Repulsion Loss.
We determined that Repulsion loss, a bounding box regression technique, could be used to improve the localization accuracy of the predicted box as shown in **Figure 1**. This paper aims to minimize the overlap of the predicted box with other ground-truths as shown in **Figure 1**, using a loss function which is same as seen above:

$$L = L\ Attr + \alpha * LRep\ GT + \beta * LRep\ Box$$

Attraction Term (L Attr): It uses SmoothL1 to narrow the gap between the predicted box and ground-truth.
Repulsion Term (LRepGT): It penalizes the prediction box for shifting to other ground truth objects. The term is meant to keep the proposal away from the neighboring non-truth ground-truth objects.
Repulsion Term (LRepBox): It tries to keep the predictions of all ground truths separated from one another.

Furthermore, **IoG** (Intersection over Ground truth) is adopted in this approach because the bounding box regression might enlarge the bounding box size, to increase the denominator in IoU thereby reducing the loss. This approach will aid in detecting occluded pedestrians, in an image as there is a reduction in the overlap of bounding boxes due to proper localization and subsequently improves the accuracy, as they do not get suppressed during NMS.

Our approach aims at using the Faster RCNN model along with Repulsion Loss to improve the accuracy of detection of pedestrians on the VOC 2007 dataset. The Pascal VOC 2007 dataset that has been used contains train/validation/test set of 9,963 images containing 24,640 annotated objects that includes 20 classes namely:

- Person: person (in our case Pedestrians)
- Animal: bird, cat, cow, dog, horse, sheep
- Vehicle: airplane, bicycle, boat, bus, car, motorbike, train
- Indoor: bottle, chair, dining table, potted plant, sofa, tv/monitor

The methodology we have implemented is Modified Fast R-CNN (Fast R-CNN + RepLoss). We use Faster RCNN because:

- Faster R-CNN is a sliding window-based approach
- It has a dedicated region proposal network followed by max-pooling and a classifier to classify the object in the proposed region.

Then, we generate PR curves and measure the area under the curve using matplotlib as well as accuracy using mAP.

5. Implementation Details

In this project, by implementing 'Repulsion Loss' with Faster RCNN as the object detector we aim at achieving more crowd robust localization. As Faster R-CNN has a dedicated region proposal network that improves the accuracy and decreases the occlusion effect faced by pedestrian detectors.

We executed the "Repulsion Loss: Detecting Pedestrians in a Crowd" for 10k images of the VOC 2007 dataset on p2.xlarge instance of AWS having Nvidia Tesla K80 GPU, PyTorch 0.4.0 and TorchVision 0.2.0. Initially, we ran the code provided in the paper to understand the overall flow as shown in the flowchart in **Figure 6**.

Training:

We implemented Repulsion Loss with Faster RCNN to obtain more optimized results. We chose training parameters like batch size, weight decay, momentum, gamma and learning rate otherwise default values are used. Training starts by loading the total number of classes including a class for the background to initialize the Faster RCNN. Next, during the forward pass, we feed image data to the base model to obtain a base feature map. Then, we feed the base feature map TP RPN to obtain ROIs(Region of Interest). We need ROI because it enriches the IMDb's ROIdb by adding some derived quantities that are useful for training. This function precomputes the maximum overlap, taken over ground-truth boxes, between each ROI and each ground-truth box. The class with maximum overlap is also recorded.

We use ground-truth boxes for refining. Then, we feed pooled features to the top model and compute the bbox offset. Later we compute object classification probability. The data loader then combines a dataset, a sampler, and provides an iterable over the given dataset. In each batch iteration, the initial learning rate is decayed at specific step intervals. A forward pass computes the losses. To obtain this loss our faster_rcnn.py program calls repulsion_loss.py, where the program returns three-loss values:
Localization loss, Repulsion loss, and Confidence loss. Final loss is a sum of Bounding box loss, Repulsion loss, and Classifier loss, each of which is returned by the repulsion loss function. This is followed by backpropagation where we update the weights on the batch of images. This forward Pass and backpropagation together comprise of the training phase and we train the model for about **2500 epochs**. Once, training is done, we now have to test and evaluate the model's performance.

Testing:

After completing the training, the next step is to evaluate the model's performance using the test dataset. The model is run on the test dataset and predictions are generated. It generates bounding boxes over its predictions. Overlap of predicted bounding boxes and ground truth boxes is then calculated using the maximum of the x, y coordinates of the top left corner of either ground truth box or predicted box and minimum of x, y coordinates of the bottom right corner of either box giving us the intersection area. The intersection over union (IoU) should be greater than the threshold value, 0.5 in this case, to qualify as a successful prediction. Depending on the prediction, we classify it in true positive(TP) or false positive(FP) and we calculate precision and recall as well as PR curves as shown in the next section.

6. Results

On implementing "Repulsion Loss" as described in the paper [1], and our implementation of Repulsion Loss with Faster RCNN, we observe that results are better when compared to Single Shot Detector implementation because Faster RCNN has a dedicated Region Proposal network followed by max-pooling and a classifier to classify the object in the proposed region. We used the mAP for comparing both models.

RepLoss + SSD: Figure 7 shows us the predictions of bounding boxes for the given input image**.** As seen in **Figure 7**, the IoU is > 0.5 for the predictions as there is a large overlap.

RepLoss + Faster RCNN: Figure 8 shows us the predictions of bounding boxes for the same input image**.** As seen in **Figure 8**, the IoU is > 0.5 for the predictions as there is a large overlap.

The ground truth is the green boxes while the blue boxes are the predictions.

PR Curves (Precision recall curves): Precision measures the ratio of true positives overall positive predictions which includes false positives. The recall is the same as the true positive rate which is the ratio of the number of positives predicted accurately.

- **RepLoss + SSD:** In **Figure 9** shows the calculated precision and recall for the 21 classes as discussed above to generate the PR curve.
- **RepLoss + Faster RCNN:** In **Figure 10** shows the calculated precision and recall for the 21 classes as discussed above to generate the PR curve.

mAP (Mean average prediction): mAP is used to predict the accuracy of an object detector. The below table represents the mAP calculated for both Repulsion Loss with SSD and Repulsion Loss with Faster RCNN. From both the figures, we can see that **RepLoss + Faster RCNN** has **better** mAP than **RepLoss + SSD**, which can be seen in the table below:

Mean Average Precision Comparison	**mAP**	**batch size**
Repulsion loss with SSD	74.58	32
Repulsion loss with Faster RCNN	79.8	4

7. Discussion and Future Scope

Our implementation of Repulsion Loss + Fast RCNN performs well for VOC datasets. We can further work on its improvement. We can utilize other regression-based loss functions such as soft margin loss and hinge embedding loss, different types of optimizers to update the weights such as Adagrad, AdaDelta, RMSProp, NAG, Adam and various types of decaying learning rates such as time-based decay and step decay and check for improvement inaccuracy of the model.

Quantization can be utilized to significantly reduce the bandwidth and storage by representing the weights and activations from 32-bit floating-point values to 8-bit integer values. Even though the space required to store the model is reduced ~4 fold but the accuracy would be reduced by a mere 5-8 %. The quantized model will be faster as integer compute is faster than float compute.

8. References

[1].RepulsionLoss: Detecting Pedestrians in a Crowd (link: https://arxiv.org/pdf/1711.07752v2.pdf)
[2].Part level CNN for Pedestrian detection (https://arxiv.org/pdf/1810.00689v1.pdf)
[3].Object as Points (link: https://arxiv.org/pdf/1904.07850v2.pdf)
[4].Center and Scale Prediction (link: https://arxiv.org/pdf/1904.02948v2.pdf)
[5].https://medium.com/overture-ai/part-5-object-detection-when-image-classification-just-doesnt-cut-it-b4072fb1a03d
[6].https://en.wikipedia.org/wiki/Object_detection
[7].https://towardsdatascience.com/learning-rate-schedules-and-adaptive-learning-rate-methods-for-deep-learning-2c8f433990d1
[8].https://medium.com/datadriveninvestor/overview-of-different-optimizers-for-neural-networks-e0ed119440c3
[9]https://medium.com/@jonathan_hui/what-do-we-learn-from-single-shot-object-detectors-ssd-yolo-fpn-focal-loss-3888677c5f4d
[10].https://medium.com/@jonathan_hui/object-detection-speed-and-accuracy-comparison-faster-r-cnn-r-fcn-ssd-and-yolo-5425656ae359
[11].https://towardsdatascience.com/breaking-down-mean-average-precision-map-ae462f623a52
[12].Faster R-CNN (link: https://arxiv.org/abs/1506.01497)

9. Appendix

Figure 1: Repulsion Loss.

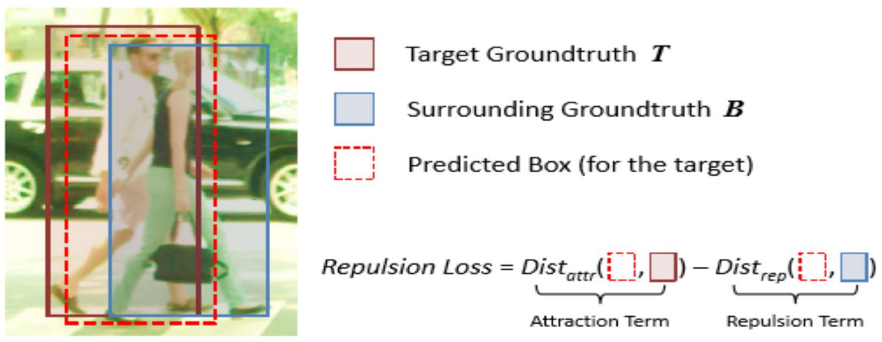

Target Groundtruth T
Surrounding Groundtruth B
Predicted Box (for the target)
Repulsion Loss = Dist_attr(,) − Dist_rep(,)
Attraction Term
Repulsion Term

Figure 2: Part Level CNN using Saliency and Boundary Box Alignment.

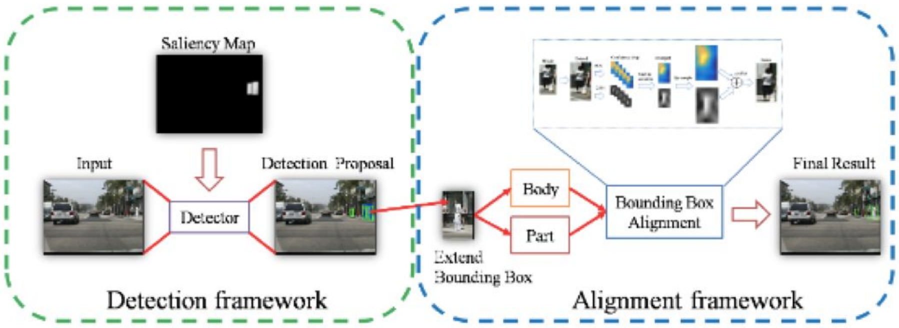


Figure 3: Object as Points.

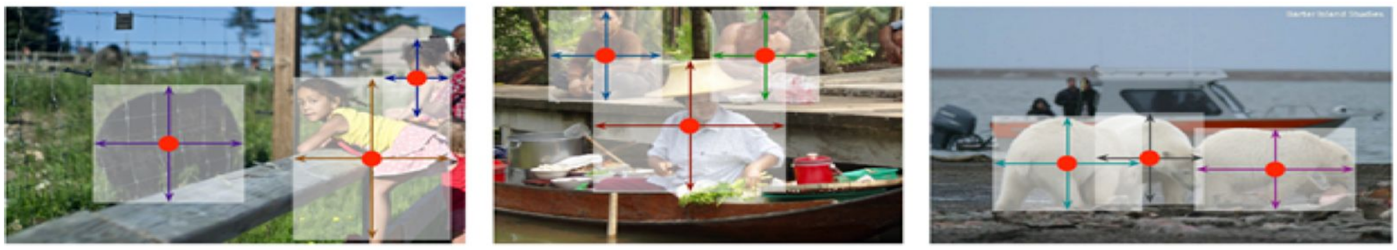


Figure 4: Center and Scale Prediction.

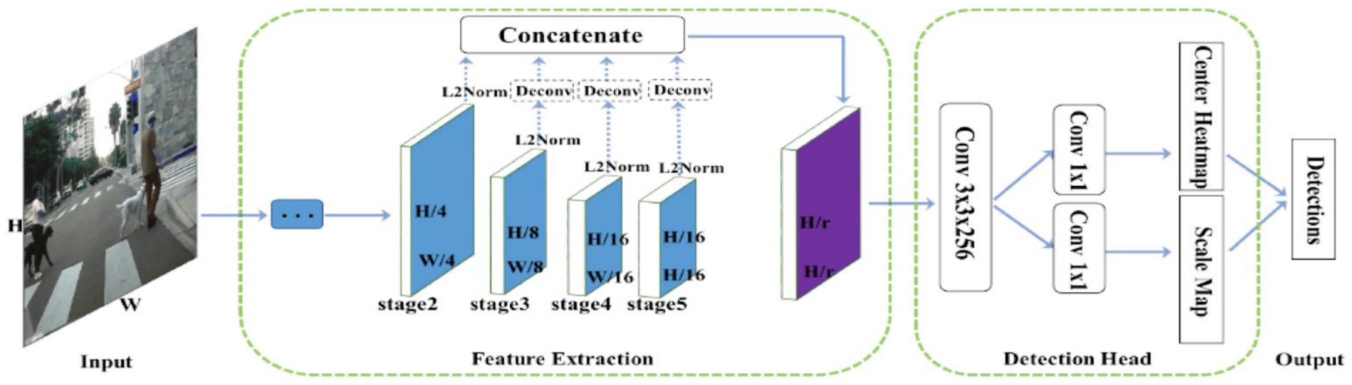
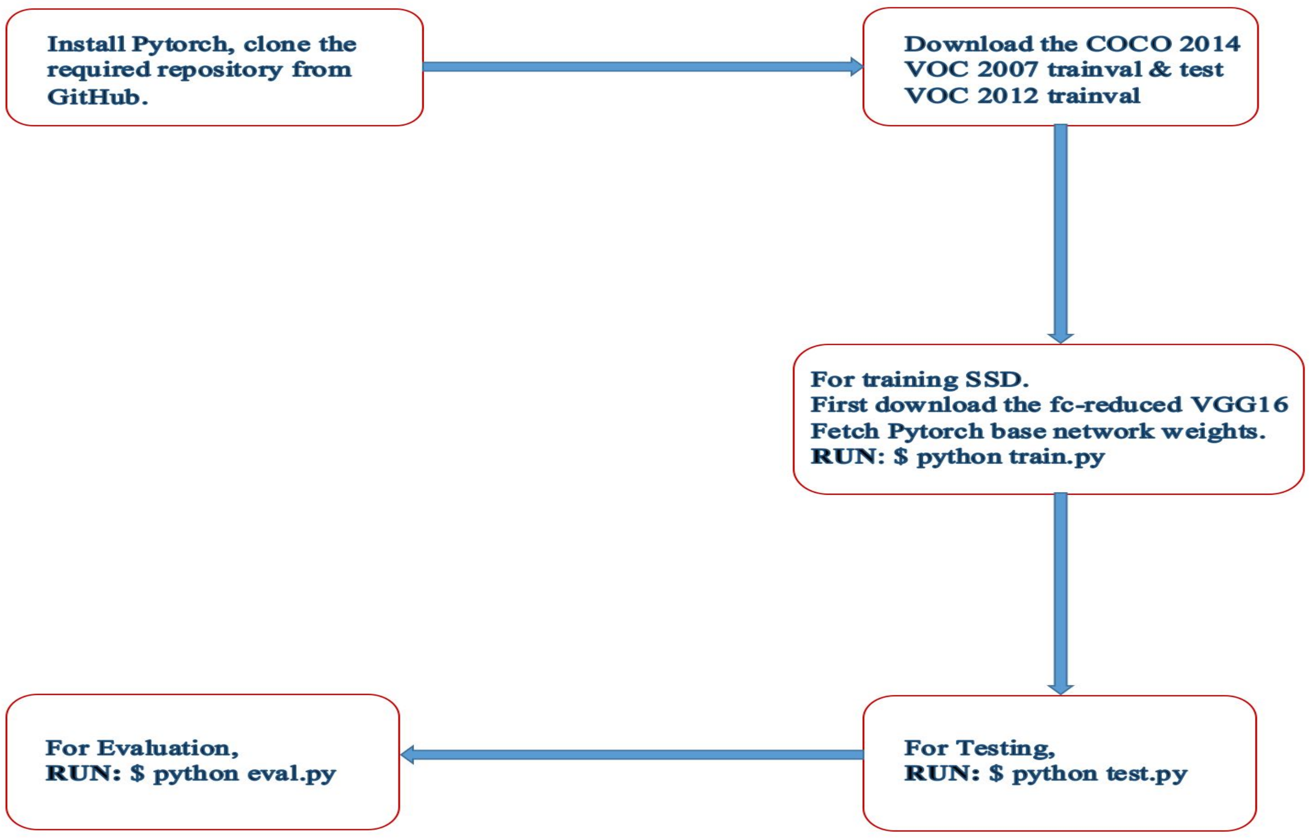


Figure 5: Flowchart for Implementation details for **RepLoss+SSD**.


Install Pytorch, clone the required repository from GitHub.

Download the COCO 2014
VOC 2007 trainval & test
VOC 2012 trainval

For training SSD.
First download the fc-reduced VGG16
Fetch Pytorch base network weights.
RUN: $ python train.py

For Testing,
RUN: $ python test.py

For Evaluation,
RUN: $ python eval.py

Figure 6: Flowchart for Implementation details for **RepLoss+Faster RCNN**.

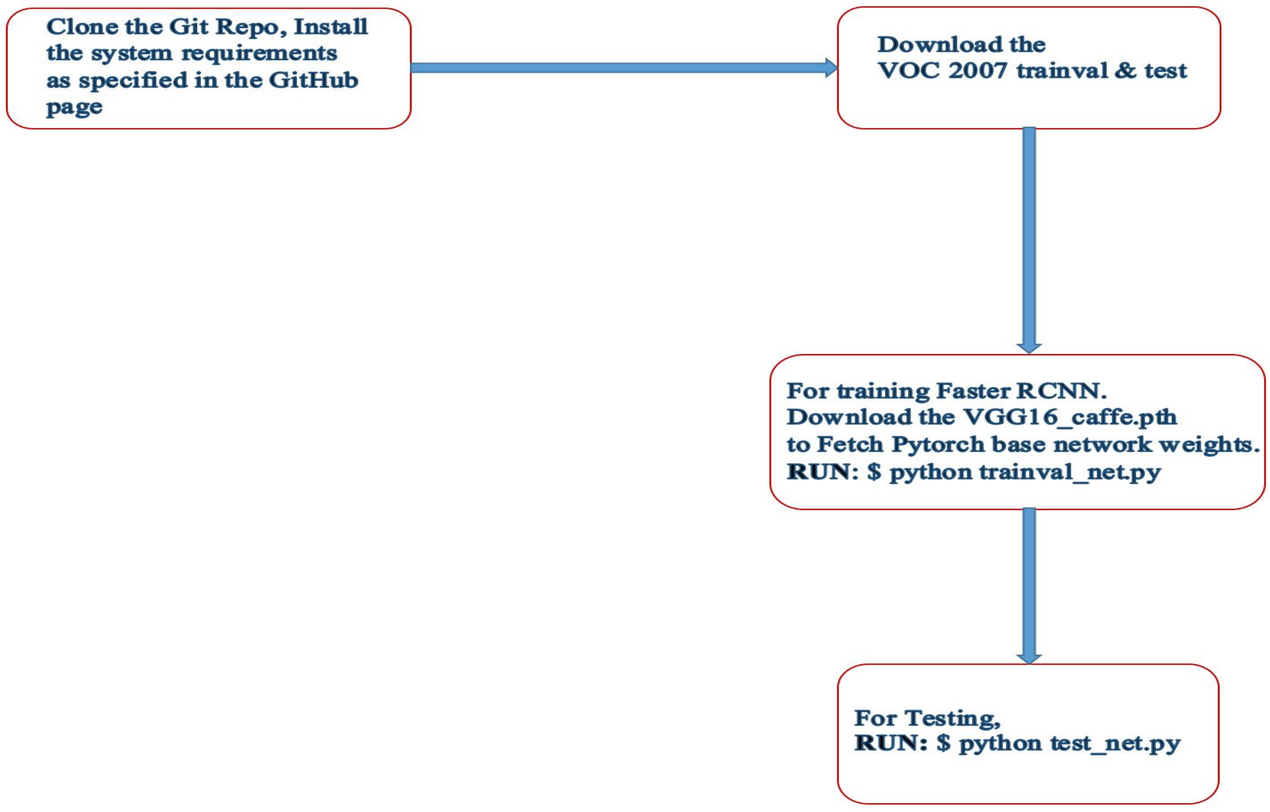

Clone the Git Repo, Install the system requirements as specified in the GitHub page
Download the VOC 2007 trainval & test
For training Faster RCNN. Download the VGG16_caffe.pth to Fetch Pytorch base network weights. RUN: $ python trainval_net.py
For Testing, RUN: $ python test_net.py

Figure 7: Ground Truth (Green) vs Predicted Bounding Box (Blue) using SSD with repulsion loss

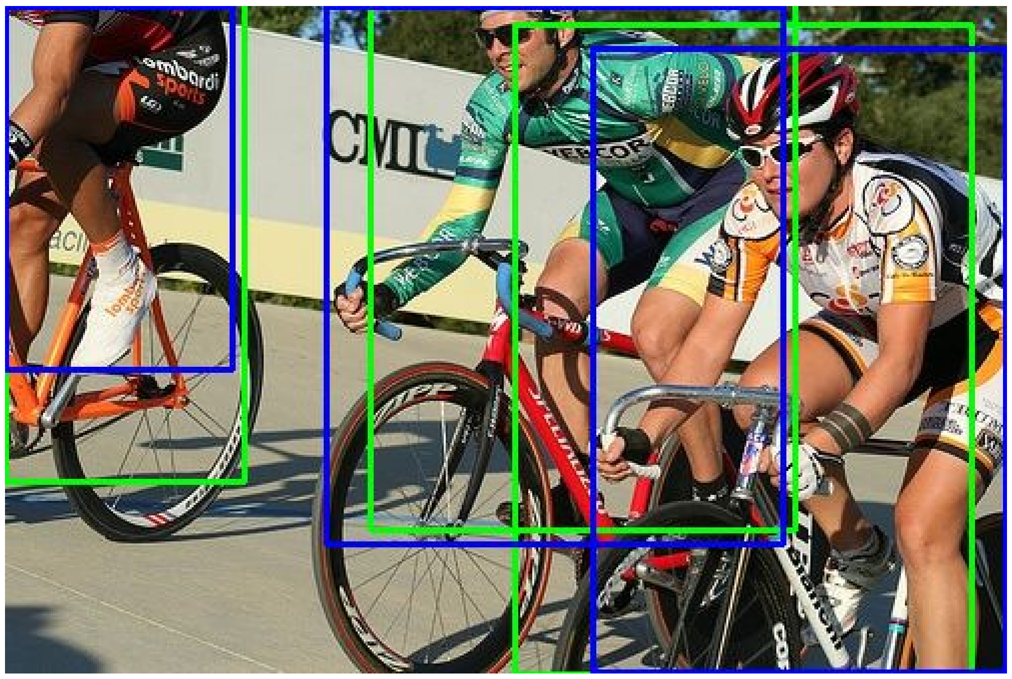


Figure 8: Ground Truth (Green) vs Predicted Bounding Box (Blue) using Faster R-CNN with repulsion loss

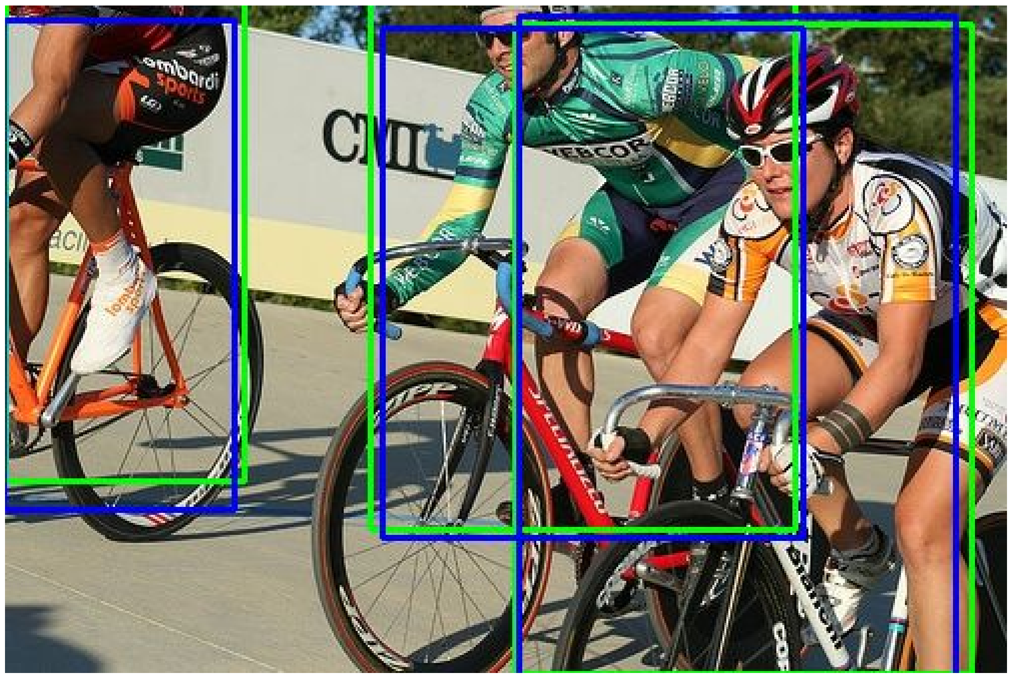


Figure 9: PR curve for repulsion loss using SSD as object detector.

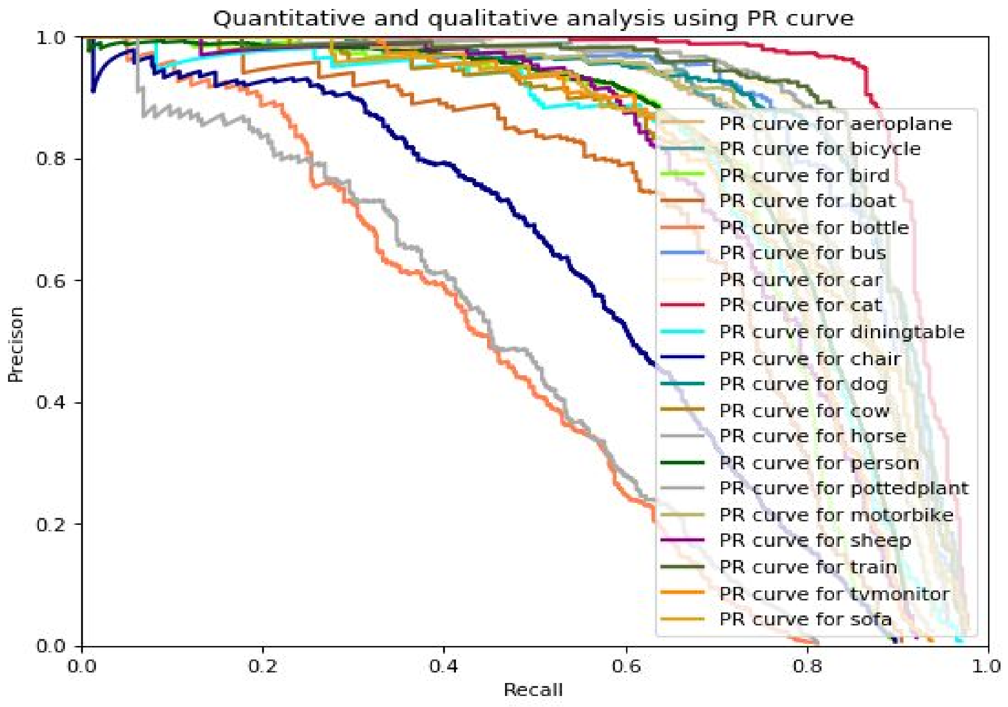
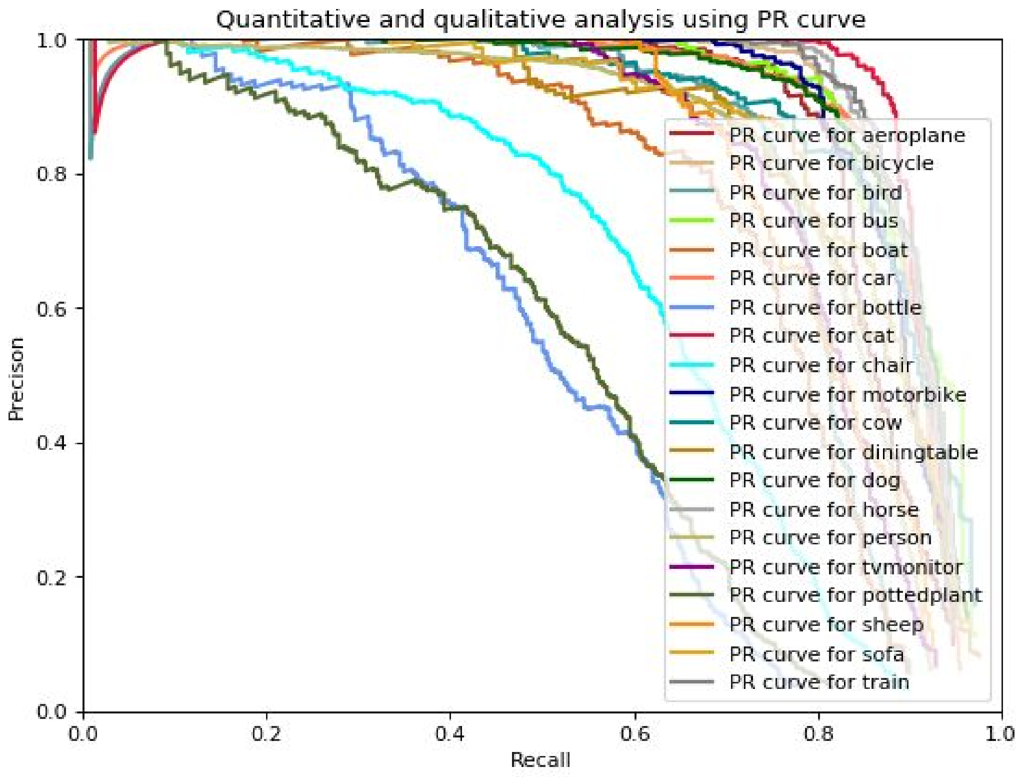


Figure 10: PR curve for repulsion loss using Faster R-CNN as object detector.


Quantitative and qualitative analysis using PR curve
Precison
Recall
PR curve for aeroplane
PR curve for bicycle
PR curve for bird
PR curve for bus
PR curve for boat
PR curve for car
PR curve for bottle
PR curve for cat
PR curve for chair
PR curve for motorbike
PR curve for cow
PR curve for diningtable
PR curve for dog
PR curve for horse
PR curve for person
PR curve for tvmonitor
PR curve for pottedplant
PR curve for sheep
PR curve for sofa
PR curve for train