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

Prof. Mashhour Solh

Milestone 1

Group B

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

Fall 2019

I. Introduction

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

II. Problems Associated with Pedestrian Detection

The core challenge lies in accurately detecting pedestrians in complex backgrounds while handling multiple issues such as crowd occlusion, body pose, lighting, deformation, scaling, etc. Many times detectors that work well on detecting humans suffer in accuracy due to occlusion. The detected bounding box encompasses multiple humans rather than generating individual bounding boxes for each human. Another issue faced by the bounding box is that they tend to leave out parts of the human body such as the forearm, lower leg, etc. This could be due to the generation of lower resolution feature maps. Furthermore, the images may contain humans at varying levels of depth causing the distant humans to seem smaller in height. This leads to a spike in the count of false positives during classification.

III. Classical Approaches to Pedestrian Detection

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.

IV. Literature Review

i. Repulsion Loss: Detecting Pedestrians in a Crowd

Available Datasets :

- **Caltech USA** dataset consists of approximately 10 hours of 640 × 480 30Hz video taken from a vehicle driving through regular traffic in an urban environment.
- **CityPersons** is a new set of person annotations on top of the Cityscapes dataset which contains a diverse collection of images from 50 cities and different seasons with 5000 finely annotated images and 20,000 coarsely annotated images.

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 $1/8^{th}$ of the input size. It also proposes a loss function: $L = L_{Attr} + \alpha * L_{Rep\ GT} + \beta * L_{Rep\ Box}$

Figure 1. Illustration of our proposed repulsion loss. The repulsion loss consists of two parts: the attraction term to narrow the gap between a proposal and its designated target, as well as the repulsion term to distance it from the surrounding non-target objects.

Attraction Term (L_{Attr}): It uses $Smooth_{L1}$ to narrow the gap between the predicted box and ground-truth.
Repulsion Term (L_{RepGT}): 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 (L_{RepBox}): It tries to keep the predictions of all ground truths separated from one another.

$$L_{\text{Attr}} = \frac{\sum_{P \in \mathcal{P}_+} \text{Smooth}_{L1}(B^P, G^P_{Attr})}{|\mathcal{P}_+|}. \qquad L_{\text{RepGT}} = \frac{\sum_{P \in \mathcal{P}_+} \text{Smooth}_{ln}\left(IoG(B^P, G^P_{Rep})\right)}{|\mathcal{P}_+|},$$

$$L_{\text{RepBox}} = \frac{\sum_{i \neq j} \text{Smooth}_{ln}\left(IoU(B^{P_i}, B^{P_j})\right)}{\sum_{i \neq j} \mathbb{1}[IoU(B^{P_i}, B^{P_j}) > 0] + \epsilon},$$

The paper uses IoU or IoG over $Smooth_{L1}$ for the repulsion terms is because both IoU and IoG are bound by the range [0, 1] whereas $Smooth_{L1}$ 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 neighbouring 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. The paper outperforms the best reported performance on two popular databases namely Caltech and CityPersons.

ii. Part-Level Convolutional Neural Networks for Pedestrian Detection Using Saliency and Boundary Box Alignment

Available Datasets :

- **Caltech USA** dataset consists of approximately 10 hours of 640 × 480 30Hz video taken from a vehicle driving through regular traffic in an urban environment.
- **INRIA** dataset consists of 1,382 training images and 288 testing images taken from a personal digital image collections or the web using Google images.
- **ETH** dataset consists of 1,450 training images and 354 testing images with a resolution of 640 × 480 (bayered). The dataset provides the camera calibration and annotations of pedestrian bounding boxes.

This paper deals with minimising the loss of body parts due to the proposal shift problem by implementing part level CNN approach having two sub networks: 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.

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 utilised. By solving the proposal shift problem, the experiment confirmed that FPPI decreases 12.40% when the bounding box alignment is applied.

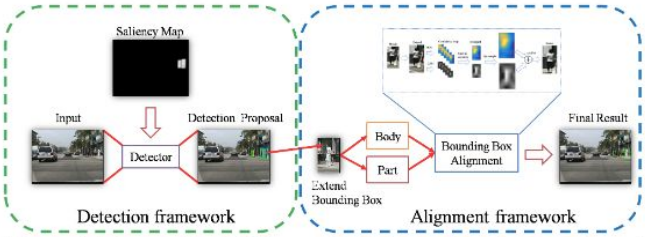


Fig. 2: Whole framework of the proposed method. The proposed pedestrian network consists of two sub-networks: detection and alignment.

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 L_d and L_S are losses of the detection network and of the saliency network, respectively.

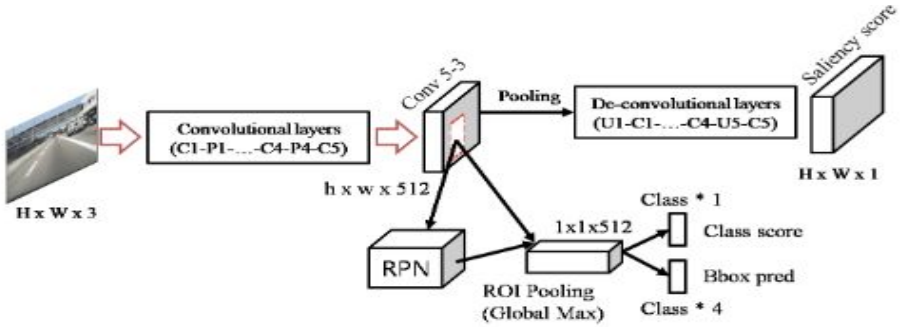


Fig. 3: Architecture of the proposed detection sub-network.

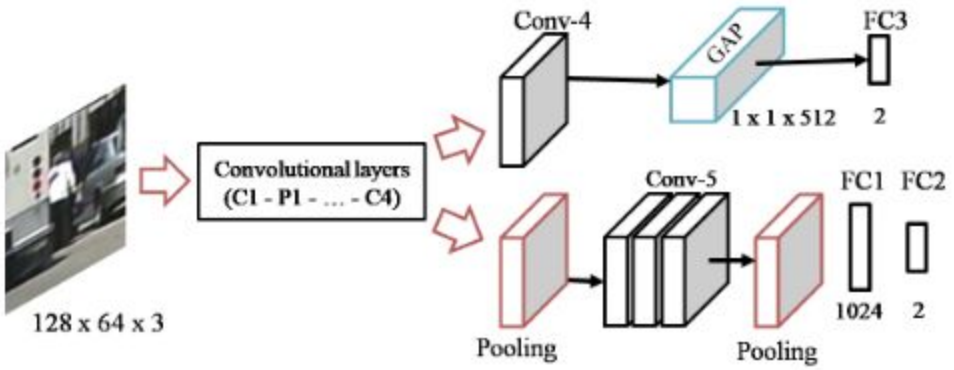
Secondly, the effective removal of false positives is done using saliency by assigning different weights to pedestrians & backgrounds. It updates the class probability (score) using the saliency map as follows:

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

where $\mathbf{f_w(b)}$ 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. $\mathbf{th_b}$ is the threshold value for distinguishing between foreground and background.

PART B: Alignment Framework

In alignment sub-network, it then successfully recalls the lost body parts by boundary box alignment using 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.



Network architecture of the proposed part-level detector based on VGG-16 network with class activation map

iii. Object as Points

Available dataset:

- **MS COCO** dataset which contains 118k training images (train2017), 5k validation images (val2017) and 20k hold-out testing images (test-dev).

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, $I \in RW \times H \times 3$, where width W and height H. Next, a key-point heat map is produced, $\hat{Y} \in [0, 1]$ W R × H R ×C, where R is the output stride and C is the number of key-point types. A prediction $\hat{Y}$ x,y,c = 1 corresponds to a detected key-point, while $\hat{Y}$ x,y,c = 0 is considered to be background. Here, several different fully convolutional encoder-decoder networks are used to predict $\hat{Y}$ 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.

At inference time, the peaks in the heat map 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.

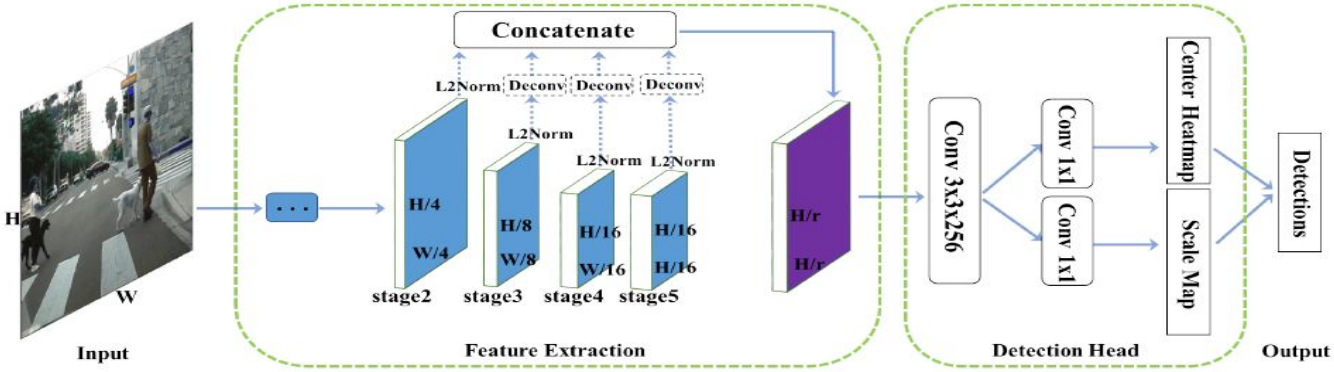
iv. Center and Scale Prediction (CSP): A Box-free Approach for Object Detection

Available Datasets :

- **Caltech USA** dataset consists of approximately 10 hours of 640 × 480 30Hz video taken from a vehicle driving through regular traffic in an urban environment.
- **Cityscapes** dataset contains a diverse collection of images from 50 cities and different seasons with 5000 finely annotated images and 20,000 coarsely annotated images.
- **CityPersons** is a new set of person annotations on top of the Cityscapes dataset.
- **Imagenet** is based on a hierarchical structure provided by WordNet having 12 subtrees with 5247 synsets and 3.2 million cleanly labeled full resolution images in total.

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 object 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, λ c, λ s, λ o are the weights with respect to each of the loss functions.

Classification Loss (L_{center}): 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

$$\hat{p}_{ij} = \begin{cases} p_{ij} & \text{if } y_{ij} = 1 \\ 1 - p_{ij} & \text{otherwise,} \end{cases}$$

and

$$\alpha_{ij} = \begin{cases} 1 & \text{if } y_{ij} = 1 \\ (1 - M_{ij})^{\beta} & \text{otherwise.} \end{cases}$$

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

The addition of the offset prediction network is optional depending on the improvement in the overall accuracy.

V. Comparison of the approaches discussed

Approaches	Methodology	Scale Variation	Occluded pedestrian detection	Accuracy
Repulsion-Loss: Detecting Pedestrians in a Crowd	Modified Fast R-CNN (Fast R-CNN + 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	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	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	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 pedestrians detection is possible using part-level detector.	The proposed method achieves comparable performance of 10.34% to state-of-the-art in a partially-occlusion INRIA dataset.

VI. Project Proposal

Through literature review, we analysed the most dominant aspects for improvement of Pedestrian Detection namely Centre-Point Estimation, Bounding Box Alignment, Center and Scale Prediction, and Repulsion Loss. While all of these have performed well on Caltech dataset, we believe that there is scope to improve the accuracy by experimenting with different layers in architecture in "*Repulsion Loss: Detecting Pedestrians in a Crowd*". Alternatively, we can implement Repulsion Loss, which is already implemented using PyTorch, by using Tensorflow

VII. 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://en.wikipedia.org/wiki/Object_detection
6. https://en.wikipedia.org/wiki/Pedestrian_detection
7. https://medium.com/adventures-with-deep-learning/focal-loss-demystified-c529277052de
8. https://www.freecodecamp.org/news/an-intuitive-guide-to-convolutional-neural-networks-260c2de0a050/