IE643: Deep Learning : Theory and Practice **Jul-Nov 2019**

End-term Project Report : Weakly Supervised Region Proposal Network and Object Detection

Student Name: Shubham Sharma *Roll No: 18i190002*

Abstract

In case of fully supervised object detection algorithms, convolution neural network plays an important role in region proposals network but in case of weakly supervised object detection algorithms, we are mainly dependent on standard region proposal networks like selective search[1]. This project is based on a paper which propose a weakly supervised region proposal network which has two stages and is dependent on only image level annotations. The first stage evaluates the objectness score of the initial bounding boxes and choose the bounding boxes with high objectness scores and the second stage refines the bounding box from the first stage using a region based CNN network

1 Introduction

Convolutional neural networks have played a huge role in fully supervised object detection methods. However, it is very laborious to collect bounding box annotations. This project concentrates on object detection with only image-level annotations, *i.e.* indicating weather an image belongs to a particular object class or not.

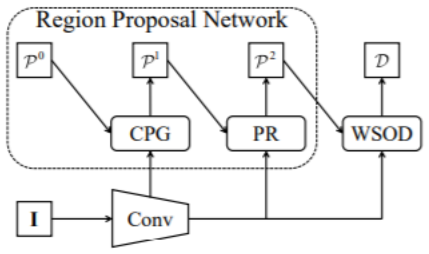


Figure 1: the overall architecture. "**I**": input image: " P^0": The initial proposal by sliding window, "$\mathcal{P}^1$: the proposals from first stage of the network, "$\mathcal{P}^2$": the proposals from second stage of the network, "$\mathcal{D}$": the detection results. "Conv": convolution layers, CGP: coarse proposal generation, "PR": proposal refinement, "WSOD": weakly supervised object detection

The project is basically divided into three stages: course proposal generation(CPG), proposal refinement(PR) and weakly supervised object detection(WSOD) in Section 2.We give details on data-set used in Section 3. We give the details of experiments and results in section 4. A description of Contributions and Related works is given in Section 5. We conclude with a short summary, pointers to future work in Section 6

2 Methods and Approach

In this section, we'll see the three main subsections and stages of the project that are :

- Course Proposal Generation(CPG)
- Proposal Refinement(PR)
- Weakly Supervised Object Detection(WSOD)

2.1 Coarse Proposal Generation

Let the initial proposals from the sliding window be $\mathcal{P}^0 = \{(b_n^0, o_n^0)\}_{n=1}^{N^0}$ of the image $\mathbf{I}$. These proposals are bounding box candidates that can be thousands or even million in number. Thus, coarse proposal generation is a step to give objectness score to each of them and then choose the candidates of bounding boxes with high scores.

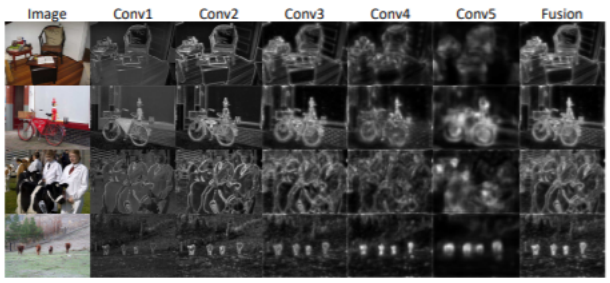


Figure 2: The responses of different convolutional layers from the VGG16[3] network trained on the ImageNet[4] data-set using only image level-level annotations. Results from left to right are the original image. responses from the first to fifth layer, and the fusion of responses from the second layer to the forth layer

CPG uses a pre-trained VGGNet[3] or other similar architecture and to refine the proposals. We take the output from from conv2 to conv4 and use it to create objectness score by Edge boxes[2]. We pass our proposals to a pre-tained VGGNet[3] and then resize the output from conv2 to conv4 to the size of the original image and take the mean of each channel to visualize the output. We are only using second to forth conv layer as they have high responses on edges and relatively low responses on other parts of the image 2. Also we are not using first and the fifth conv layer as the former has high responses on most of the image regions and the later tends to fire on the whole object instead of edges. We then take the average of these resized outputs and use that to give objectness scores to our proposals and then create $\mathcal{P}^1 = \{(b_n^1, o_n^1)\}_{n=1}^{N^1}$ and apply NMS(non-maximum suppression), which is input to our next step.

2.2 Proposal refinement

We have our proposals from CPR $\mathcal{P}^1 = \{(b_n^1, o_n^1)\}_{n=1}^{N^1}$. We use region-based CNN classifier to re-evaluate the objectness score $\tilde{o}_n^1$. We evaluate $\tilde{o}_n^1 = h(o_n^1, f(\mathbf{I}, b_n^1)) = o_n^1.f(\mathbf{I}, b_n^1)$ to reject the proposals with low scores. To do this, we first extract the conv features from b_n^1 and resize it to $512 \times 3 \times 3$ using RoI pooling method. Then we pas sit to two 256-dimension Fully connected(FC) layers to obtain the object proposal feature vector. Finally a softmax layer is used to distinguish whether the proposal is object or background. Accordingly, we obtain proposals $\tilde{\mathcal{P}}^1 = \{(b_n^1, \tilde{o}_n^1)\}_{n=1}^{N^1}$ with re-evaluated objectness score $\tilde{o}_n^1$.

To get the final score, we can simply rank the proposals according to the objectness score $\tilde{o}_n^1$ and select some proposals with top objectness scores. But there might be many redundant proposals(*i.e.* highly overlapped proposals) in $\tilde{\mathcal{P}}^1$. Therefore, we apply NMS on $\tilde{\mathcal{P}}^1$ and keep N^2 proposals with highest objectness scores. Accordingly, we obtain $\mathcal{P}^2 = \{(b_n^2, o_n^2)\}_{n=1}^{N^2}$

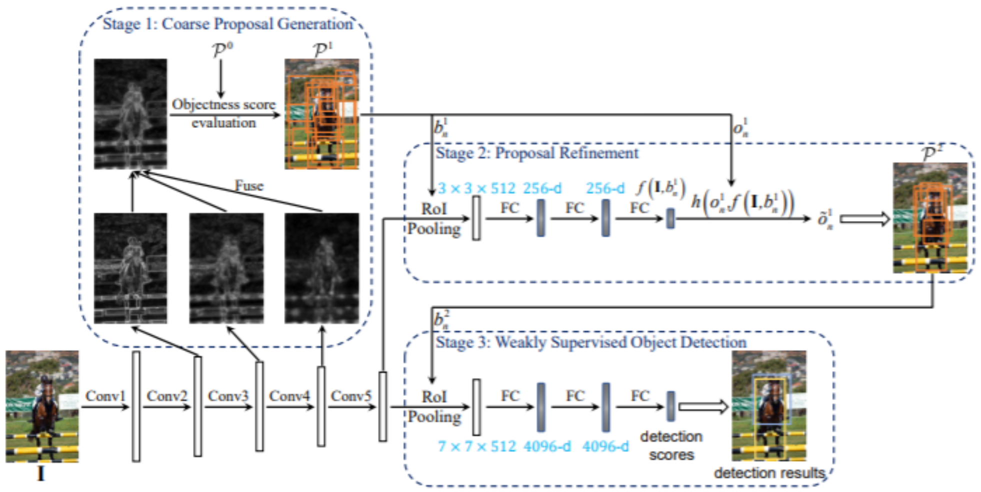


Figure 3: The detailed architecture of the network

2.3 Weakly Supervised Object Detection

We now have proposals $\mathcal{P}^2 = \{(b_n^2, o_n^2)\}_{n=1}^{N^2}$ and thus weakly supervised object detection(WSOD) classifies proposals $\mathcal{P}^2$ into different object classes. Given the proposals $\mathcal{P}^2$, for each of the proposal, we take the convolution map of the proposal and pass it to RoI pooling method to get a $512 \times 7 \times 7$ feature map followed by two 4096-dimension FC layers. Then a $\{K + 1\}$-dimension FC layer is used to which classify the b_n^2 into K object classes or background. Finally NMS is used to remove redundant detection boxes and produce object detection results.

3 Data-set

The data-set that we are using in this project is **Google Open Images Dataset V4**. The images are very diverse and often contain complex scenes with several objects (8.4 per image on average) and the data-set is annotated with image-level labels spanning thousands of classes. We have taken a subset of data-set for the simplicity.
The data-set that has been used in the project has having images from only three categories: car, phone, person, with each of the category having 100 image each. The data is given in csv format with the link to the corresponding images and annotation for bounding boxes. An example of image from the data-set is shown in the figure 4

4 Work done/experiments and results

Now that we have data-set, let us see the how it is being executed step by step:

1. **Step 1: Generation of bounding boxes for $\mathcal{P}^0$**

   - The step 1 of the project is to generate random and many bounding boxes in an image that are

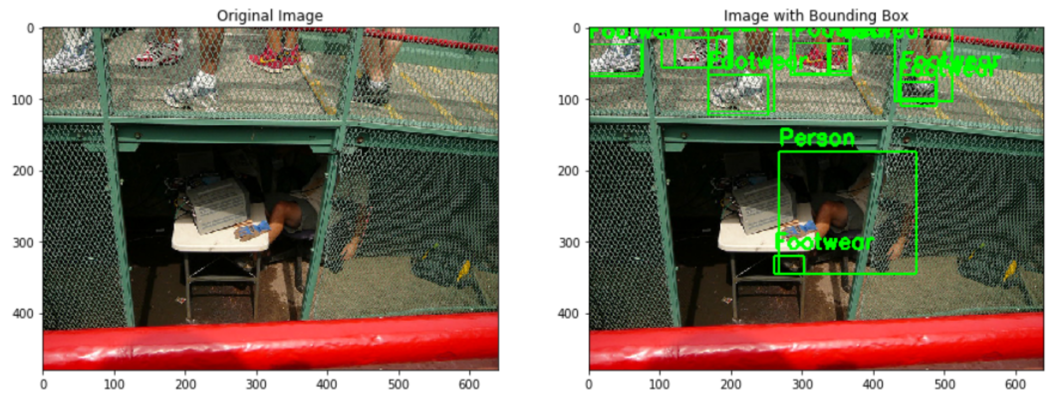


Figure 4: An image from the data-set with the corresponding detection results

the candidates for $\mathcal{P}^0$. We have used selective search for this and example of the proposals $\mathcal{P}^0$ is i the figure below:

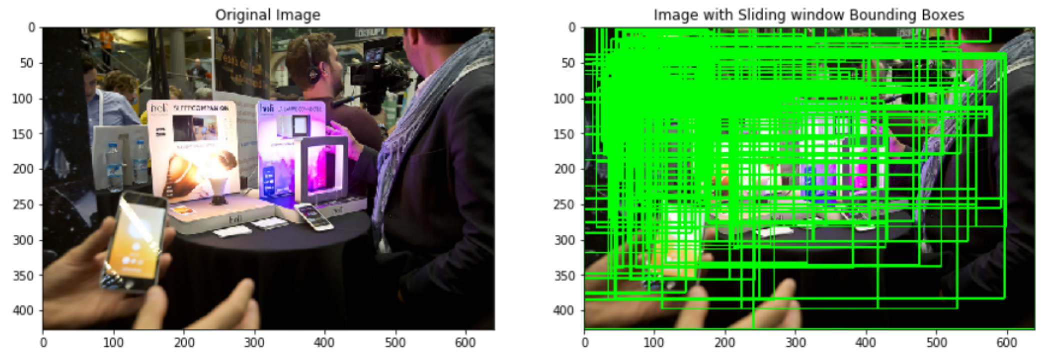


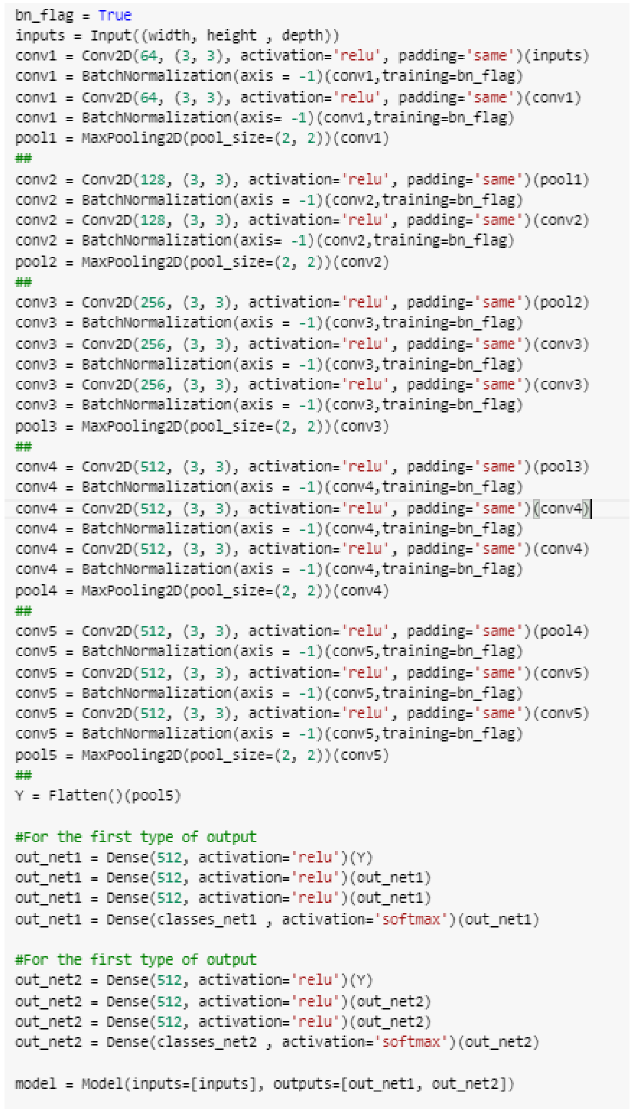
Figure 5: The sliding window results from a random image

2. **Step 2: Designing of the network**

   - We now have to make a network in such a way that it classifies whether a region is foreground or background. It should also classify among different types of object and none of them. For this, the network that we have designed(different from the research paper) accepts as input regions from the image and give the two outputs. As we can see from the figure below, the flattened layer Y is given as input to two different paths out_net1 resulting in prediction of being an object or not and out_net2 resulting in prediction of being one of different objects or none of them. The network has been trained in google colab GPU with adam optimizer and lr rate = 1e-2.

3. **Step 3: Preparation of the data-set for the network**

   - Now that the architecture of the network has been made, we'll have two prepare the data. We have images of persons, phone, car having 100 of each. We have extracted the regions with objects from all of these images with 224 car images, 117 phone images, 281 person images and hence 622 object images. Also the none of these images are 269 in number. Corresponding to these we have made the corresponding array of labels and are saved in npy format.



```
bn_flag = True
inputs = Input((width, height , depth))
conv1 = Conv2D(64, (3, 3), activation='relu', padding='same')(inputs)
conv1 = BatchNormalization(axis = -1)(conv1,training=bn_flag)
conv1 = Conv2D(64, (3, 3), activation='relu', padding='same')(conv1)
conv1 = BatchNormalization(axis= -1)(conv1,training=bn_flag)
pool1 = MaxPooling2D(pool_size=(2, 2))(conv1)
##
conv2 = Conv2D(128, (3, 3), activation='relu', padding='same')(pool1)
conv2 = BatchNormalization(axis = -1)(conv2,training=bn_flag)
conv2 = Conv2D(128, (3, 3), activation='relu', padding='same')(conv2)
conv2 = BatchNormalization(axis= -1)(conv2,training=bn_flag)
pool2 = MaxPooling2D(pool_size=(2, 2))(conv2)
##
conv3 = Conv2D(256, (3, 3), activation='relu', padding='same')(pool2)
conv3 = BatchNormalization(axis = -1)(conv3,training=bn_flag)
conv3 = Conv2D(256, (3, 3), activation='relu', padding='same')(conv3)
conv3 = BatchNormalization(axis = -1)(conv3,training=bn_flag)
conv3 = Conv2D(256, (3, 3), activation='relu', padding='same')(conv3)
conv3 = BatchNormalization(axis = -1)(conv3,training=bn_flag)
pool3 = MaxPooling2D(pool_size=(2, 2))(conv3)
##
conv4 = Conv2D(512, (3, 3), activation='relu', padding='same')(pool3)
conv4 = BatchNormalization(axis = -1)(conv4,training=bn_flag)
conv4 = Conv2D(512, (3, 3), activation='relu', padding='same')(conv4)
conv4 = BatchNormalization(axis = -1)(conv4,training=bn_flag)
conv4 = Conv2D(512, (3, 3), activation='relu', padding='same')(conv4)
conv4 = BatchNormalization(axis = -1)(conv4,training=bn_flag)
pool4 = MaxPooling2D(pool_size=(2, 2))(conv4)
##
conv5 = Conv2D(512, (3, 3), activation='relu', padding='same')(pool4)
conv5 = BatchNormalization(axis = -1)(conv5,training=bn_flag)
conv5 = Conv2D(512, (3, 3), activation='relu', padding='same')(conv5)
conv5 = BatchNormalization(axis = -1)(conv5,training=bn_flag)
conv5 = Conv2D(512, (3, 3), activation='relu', padding='same')(conv5)
conv5 = BatchNormalization(axis = -1)(conv5,training=bn_flag)
pool5 = MaxPooling2D(pool_size=(2, 2))(conv5)
##
Y = Flatten()(pool5)

#For the first type of output
out_net1 = Dense(512, activation='relu')(Y)
out_net1 = Dense(512, activation='relu')(out_net1)
out_net1 = Dense(512, activation='relu')(out_net1)
out_net1 = Dense(classes_net1 , activation='softmax')(out_net1)

#For the first type of output
out_net2 = Dense(512, activation='relu')(Y)
out_net2 = Dense(512, activation='relu')(out_net2)
out_net2 = Dense(512, activation='relu')(out_net2)
out_net2 = Dense(classes_net2 , activation='softmax')(out_net2)

model = Model(inputs=[inputs], outputs=[out_net1, out_net2])
```

Figure 6: The architecture of the network designed

4. **Step 4: Generating $\mathcal{P}^1$ from $\mathcal{P}^0$**

    - The network that was defined above is trained and the saved weights are saved. The corresponding weights are used in further prediction. Now, in order to generate $\mathcal{P}^1$ from $\mathcal{P}^0$, we pass all the region proposals in the classifier and take the results from conv2, conv3 and conv4 as defined earlier and put a threshold by taking the objectness score in order to get $\mathcal{P}^1$. An example of the output that we have got is:

5. **Step 5: Generating $\mathcal{P}^2$ from $\mathcal{P}^1$**

    - The generation of $\mathcal{P}^2$ from $\mathcal{P}^1$ includes the out_net1 part of the network in which we use the probability of the object being an object from the network that we have designed.

6. **Step 6: Weakly Supervised Object Detection**

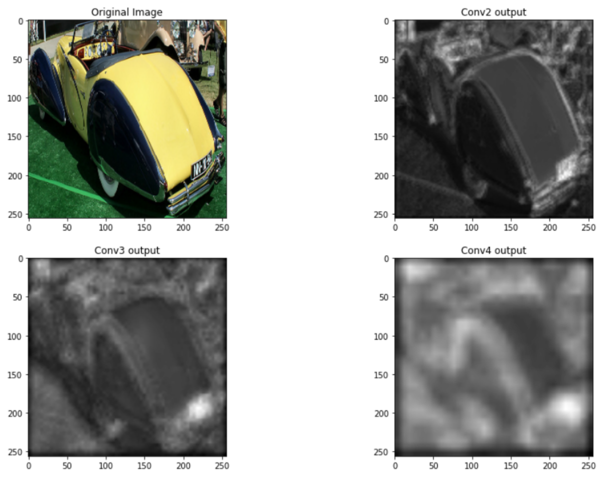


Figure 7: Outputs from the convolutional layers

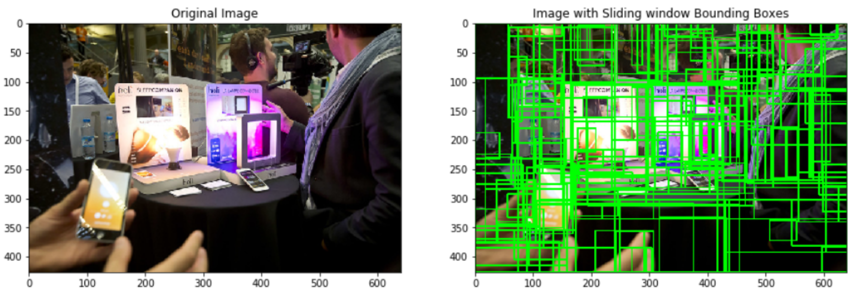


Figure 8: Proposals P^1 for an image

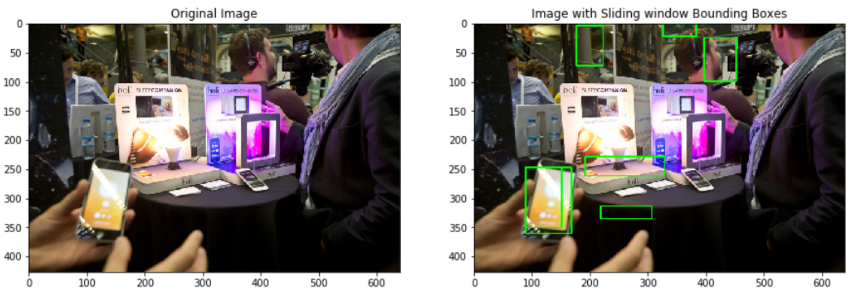


Figure 9: Proposals P^2 for an image

- The last step is for detecting the objects from the proposals $\mathcal{P}^2$. the proposals are again passed through the network and the out_net2 are considered for the object detection.

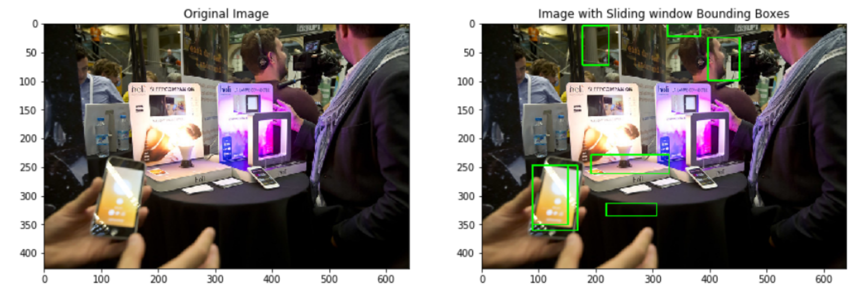


Figure 10: Output of WSOD

5 Contributions and Related works

- The research paper already had a pre-trained network to classify the objectness and the type of object but we have made a network as that network was unavailable
- A single network has been made to do all the three stages of the project
- Some of the other related works that are read in this project are Faster r-cnn, VGGnet, Selective Search, Fast r-cnn.

6 Conclusion

We have seen that with the help of image-level annotations, we can do object detection. This project has maily focused step by step methods used to give proposals. Region proposal generation contains two stages: coarse proposal generation(CPR) and proposal refinement(PR). We have also used a region based CNN[5] in PR. These steps are then followed b a weakly supervised object detection, which contributed it to be three-stage network.

In the future, we can try to use image processing techniques like canny edge detection in CPR(by tuning the threshold values in canny edge detection technique). Also, the network that we have made can be trained better with a slightly more images of the none type as the network is some times not able to predict the none object as none as we have seen from the results.

References

[1] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[2] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[3] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[4] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013.

[5] C Lawrence Zitnick and Piotr Dollár. Edge boxes: Locating object proposals from edges. In *European conference on computer vision*, pages 391–405. Springer, 2014.