International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 03 | Mar 2020 www.irjet.net p-ISSN: 2395-0072

Explicit Content Detection using Faster R-CNN and SSD MobileNet v2

Animesh Srivastava[1], Anuj Dalvi[2], Cyrus Britto[3], Harshit Rai[4], Kavita Shelke[5]

[1,2,3]Computer Engineering, Fr.C Rodrigues Institute Of Technology, Navi Mumbai, India
[4]Fr.C Rodrigues Institute Of Technology, Navi Mumbai, India
[5]Assistant Professor, Dept. of Computer Engineering, Fr.C Rodrigues Institute Of Technology, Maharashtra, India

---

Abstract - *With the increase in the availability of the internet, people are consuming more and more content on a regular basis and hence the chances of getting exposed to explicit content increase exponentially. To detect this explicit content from images or videos we make use of Convolutional Neural Networks (CNNs). This paper highlights the trade-off between speed, accuracy and training methodology for explicit content detection using Faster R-CNN and SSD MobileNet v2. FRCNNs introduce Regional Proposal Networks (RPNs) replacing the Search selective process thus making it faster for object detection. On the other hand, the combination of the MobileNet v2 architecture and the Single Shot Detector(SSD) framework yield an efficient object detection model making use of depth wise separable convolution.*

Key Words: Convolutional Neural Networks, Regional Proposal Networks, Single Shot Detector

1. INTRODUCTION

With easy availability of gadgets, use of superior technology, rising internet speeds, access to enormous amounts of content has become possible. Internet has changed our lives in numerous ways. With just one search one can get a plethora of information on any topic. This makes tasks such as research, self-learning extremely smooth and straightforward. However, such easy and unmoderated access to the internet may come at a cost. The web is an environment during which individuals are susceptible to cyber bullying, fraud, loss of privacy, most importantly, exposure to explicit and abusive content directly or indirectly. This is all the more concerning when the individual consuming such content is a child.

With the advent of computing infrastructure, Deep Neural Networks [1] have achieved massive success in the domain of Computer Vision to handle such massive amounts of data. The success of AlexNet [2] in the ImageNet Large Scale visual recognition challenge [3] made Convolutional Neural Networks (CNNs), a class of deep neural networks the absolute standard for analyzing visual imagery. This in a way solved the fundamental problems like image classification [4], semantic segmentation [5], and object detection [6].

In this paper we compare Faster R-CNN [7] and SSD MobileNet v2 [8], both object detection models to detect explicit content from an image in terms of speed, accuracy and model size. Instead of using selective search algorithm, used the in slower and time-consuming Fast R-CNN [9], on the feature map to identify the region proposals, a separate network is used to predict the region proposals. The predicted region proposals are then reshaped using a RoI pooling layer which is then used to classify the image within the proposed region and predict the offset values for the bounding boxes, making Faster R-CNNs ideal for real-time object detection.

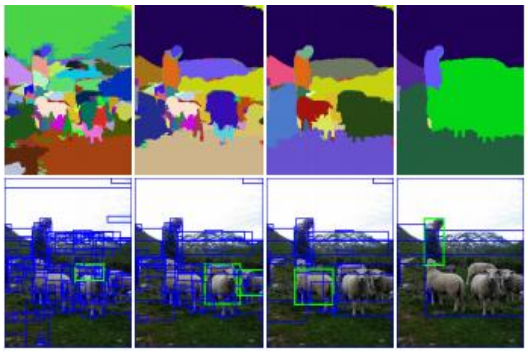

Fig -1: Search selective process used in Fast R-CNNs showing many objects at different scales [10]

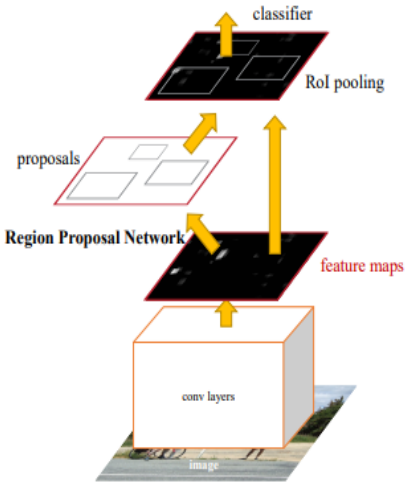

Fig -2: Faster R-CNN is a single, unified network for object detection [17]

Single Shot Detectors (SSDs) [11] originally developed by Google are a balance between Faster R-CNNs and You Only Look Once (YOLO) [12] frameworks. The SSD framework is more straightforward requiring an input image and ground truth boxes for each object during training. This approach based on a feed forward convolutional network produces a fixed-size collection of bounding boxes and scores for the presence of object class instances in those boxes, followed by a non-maximum suppression step to produce the final detections.

When building object detection networks we normally use existing network architecture, such as VGG or ResNet, and then use it inside the object detection pipeline. The problem is that these network architectures can be very large in the order of 200-500MB. Network architectures such as these are unsuitable for resource constrained devices due to their sheer size and resulting number of computations. To counter this problem we use MobileNets [13], called "MobileNets" because they are designed for resource constrained devices such as your smartphone. The general idea behind depth wise separable convolution is to split convolution into two stages:

1) 3 x 3 depth wise convolution

2) 1 x 1 point wise convolution

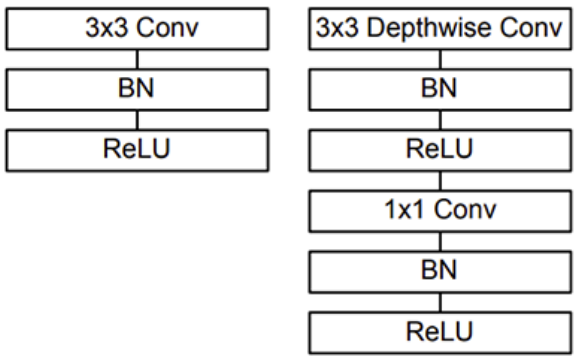

Fig -3: (Left) Standard convolutional layer with batch normalization and ReLU. (Right) Depth wise separable convolution with depth wise and point wise layers followed by batch normalization and ReLU [13]

This paper is organised as follows. Section II consists of the literature survey in the domain of Object Detection. Section III describes the approaches for dataset creation and ground truth generation. Section IV describes the comparison between pipelines of our proposed systems. Section V provides the evaluation of the work. Section VI states the inferences of our comparison and Section VI outlines our conclusion.

2. RELATED WORKS

2.1 Explicit Content Detection System: An Approach towards a Safe and Ethical Environment

Ali Qamar Bhatti et al. [14] proposed an explicit content detection (ECD) system to detect Not Suitable For Work (NSFW) media (i.e., image/ video) content. The proposed ECD system is based on residual network (i.e., deep learning model) which returns a probability to indicate the explicitness in media content. The value is further compared with a defend threshold to decide whether the content is explicit or no explicit.

The proposed system not only differentiates between explicit/no explicit contents but also indicates the degree of explicitness in any media content, i.e., high, medium, or low. In addition, the system also identifies the media fles with tampered extension and labels them as suspicious.

2.2 Pornography Detection Using Support Vector Machine

Yu-Chun Lin et al. [15] proposed an easy scheme for detecting pornography. They exploit primitive information from pornography and use this knowledge for determining whether a given photo belongs to pornography or not. They extract skin region from photos, and find out the correlation in skin region and non-skin region. Then, we use these correlations as the input of support vector machine (SVM) for classification.

2.3 An Algorithm for Nudity Detection

Rigan Ap-apid et al. [16] proposed an algorithm for detecting nudity in color images. A skin color distribution model based on the RGB, Normalized RGB, and HSV color spaces is constructed using correlation and linear regression. The skin color model is used to identify and locate skin regions in an image.

These regions are analyzed for clues indicating nudity or non-nudity such as their sizes and relative distances from each other. Based on these clues and the percentage of skin in the image, an image is classified nude or non-nude.

2.4 An algorithm of pornographic image detection

Hong Zhu. et al [17] proposed a skin model based on the combination of YIQ, YUV, and HSV. In the step of the pre-dealing, they used white balance algorithm to achieve better skin area. Then, texture model based on Gray Level Co-Matrix (GLCM) and geometric structure of human beings were used to decrease the disruptions of the background region similar with the skin area.

The features which were extracted from the last images dealt by color and texture model were input into Support Vector Machines (SVM), through which the pornographic images were classified successfully.

3. DATASET

3.1 Dataset Collection

The dataset consists of 500 images from 5 explicit classes, i.e, 100 images of each class. The classes are breasts, vagina, penis, kissing and buttocks. The images in the dataset were generated by crawling the internet. Out of the 100 images from each class, 80 were used for training and 20 were used for testing, i.e, 400 images were train images and 100 were test images. The images consisted of one or more naked human subjects both male and female. Also the images consisted of subjects with different skin colors, body hair, lighting conditions, intensity, percentage of body exposed and grayscale images.

This presents a significant additional challenge as convolutional neural networks can be expected to perform best, in general, when the input data is as uniform and standardized as possible. This includes standardization in terms of color, contrast, scale, and class balance.

All the training was performed on our local machine with a Ryzen 3 1200 CPU, Nvidia GTX 1050ti GPU and 8GB of DDR4 RAM.

3.2 Ground Truth Generation

Preparing the data involved manually annotating the image, i.e creating bounding boxes around the object to be detected using the labelImg image annotator tool [18]. The tool can be downloaded from its github page and follows a simple installation.

This tool is used to define regions in an image and create textual descriptions of those regions. Annotations are saved as XML files in PASCAL VOC format, the format used by ImageNet. Besides, it also supports YOLO format. An illustration of this tool is given in Figure 4.

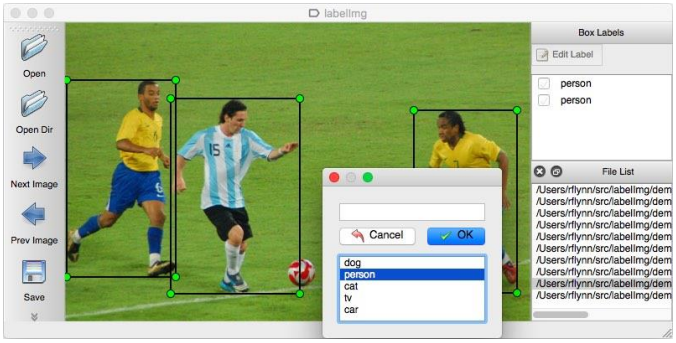

Fig -4: Manually annotated image using labelImg

4. Comparison Between Faster R-CNN and SSD MobileNet v2

4.1 Faster R-CNN

Faster R-CNN is composed of two modules. The first module is a deep fully convolutional network that proposes regions, and the second module is the Fast R-CNN detector that uses the proposed regions. The entire system is a single, unified network for object detection (Figure 2). Using the recently popular terminology of neural networks with 'attention' mechanisms, the RPN module tells the Fast R-CNN module where to look.

4.1.1 Backbone Network

This is a standard Convolutional neural network (CNN) typically ResNet50 or ResNet101 [19] which serves as the feature extractor. The lower level of the network detects features such as edges, corners whereas the later level detects higher level of features. ResNet101 is utilized as its additional layers help in leveraging the sparse dataset.

4.1.2 Region Proposal Networks

This is a fully Convolutional network which takes as input the image features from the backbone network and scans the image in a sliding window fashion to find the areas which contain objects. The areas which are scanned by the RPN are called Anchors. Then, the top anchors based on the prediction from the RPN are selected. In case the anchors overlap too much, then Non-Maximum Suppression (NMS) on the proposal regions based on their

class scores in order to reduce redundancy is adopted. The final proposal is passed onto the next step.

4.1.3 Anchors

At each sliding-window location, we simultaneously predict multiple region proposals, where the number of maximum possible proposals for each location is denoted as k. So the reg layer has 4k outputs encoding the coordinates of k boxes, and the cls layer outputs 2k scores that estimate probability of object or not object for each proposal. The k proposals are parameterized relative to k reference boxes, which we call anchors. An anchor is cantered at the sliding window in question, and is associated with a scale and aspect ratio (Figure 3). By default we use 3 scales and 3 aspect ratios, yielding k = 9 anchors at each sliding position. For a convolutional feature map of a size W × H (typically ~2,400), there are W Hk anchors in total.

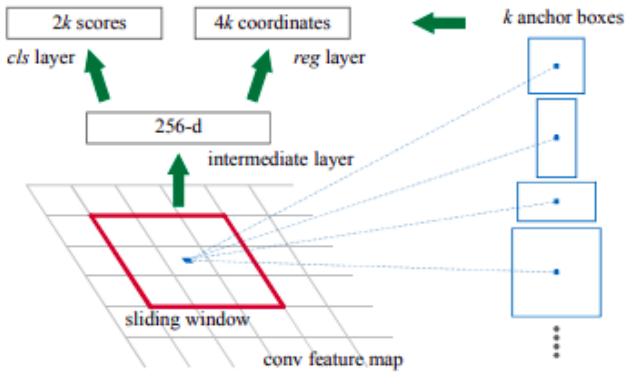

Fig -5: Region Proposal Network (RPN)

4.1.4 ROI Pooling

After RPN, we get proposed regions with different sizes. Different sized regions means different sized CNN feature maps. It's not easy to make an efficient structure to work on features with different sizes. Region of Interest Pooling can simplify the problem by reducing the feature maps into the same size. Unlike Max-Pooling which has a fix size, ROI Pooling splits the input feature map into a fixed number (let's say k) of roughly equal regions, and then apply Max-Pooling on every region. Therefore the output of ROI Pooling is always k regardless the size of input.

4.2 SSD MobileNet v2

Our SSD MobileNet v2 is pretrained on the COCO dataset [20].

4.2.1 Single Shot Detector(SSD)

The SSD approach is based on a feed-forward convolutional network that produces a fixed-size collection of bounding boxes and scores for the presence of object class instances in those boxes, followed by a non-maximum suppression step to produce the final detections. The early network layers are based on a standard architecture used for high quality image classification (truncated before any classification layers). The auxiliary structure to the network to produces detections with the following key features:

4.2.1.1 Multi-scale feature maps for detection

Convolutional feature layers are added to the end of the truncated base network. These layers decrease in size progressively and allow predictions of detections at multiple scales. The convolutional model for predicting detections is different for each feature layer.

4.2.1.2 Convolutional predictors for detection

Each added feature layer (or optionally an existing feature layer from the base network) can produce a fixed set of detection predictions using a set of convolutional filters. These are indicated on top of the SSD network architecture in Fig. 6. For a feature layer of size m × n with p channels, the basic element for predicting parameters of a potential detection is a 3 × 3 × p small kernel that produces either a score for a category, or a shape offset relative to the default box coordinates. At each of the m × n locations where the kernel is applied, it produces an output value. The bounding box offset output values are measured relative to a default box position relative to each feature map location.

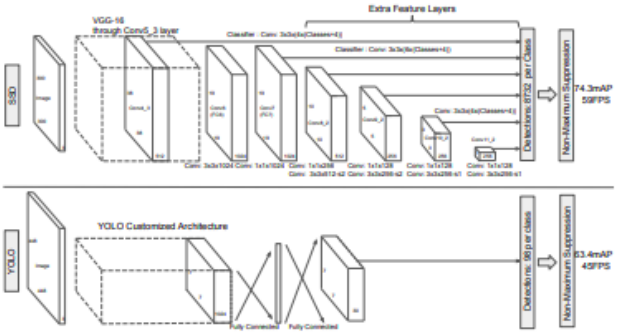

Fig -6: A comparison between two single shot detection models: SSD and YOLO [11]

4.2.1.3 Default boxes and aspect ratios

A set of default bounding boxes are associated with each feature map cell, for multiple feature maps at the top of the network. The default boxes tile the feature map in a convolutional manner, so that the position of each box relative to its corresponding cell is fixed. At each feature map cell, the offsets are predicted relative to the default box shapes in the cell, as well as the per-class scores that indicate the presence of a class instance in each of those boxes. Specifically, for each box out of k at a given location, c class scores and the 4 offsets relative to the original default box shape are computed. This results in a total of (c + 4)k filters that are applied around each location in the feature map, yielding (c + 4)kmn outputs for a m × n feature map. The default boxes are similar ton the anchor boxes used in Faster R-CNN, however they are applied to several feature maps of different resolutions. Allowing different default box shapes in several feature maps efficiently discretizes the space of possible output box shapes.

4.2.2 MobileNet v2

The basic building block is a bottleneck depth-separable convolution with residuals. ReLU6 is used as the non-linearity because of its robustness when used with low-precision computation [21]. Kernel size 3 × 3 is standard for modern networks, and utilizes dropout and batch normalization during training. This MobileNet v2 is used a feature extractor for the SSD.

5. RESULTS

Training and testing of both models for explicit content detection was performed on the Nvidia GeForce 1050 Ti which has a memory size of 4GB to observe the performance in terms of detection accuracy and speed.

5.1 Faster R-CNN

The implementation of the Faster R-CNN is done in Tensorflow and it utilizes Inception-ResNet [22] as a backbone. Due to the very less amount of Training data, transfer learning was deemed as a viable option in which the pretrained weights of the MS-COCO dataset [20] are used as a starting point for training. We trained the model for 150k steps with initial learning rate as 0.0002 and batch size 1.

Table -1: Inference for Faster R-CNN on images

Model	Inference Speed	COCO mAP
Faster R-CNN	15ms	28

Table -2: Inference for Faster R-CNN on videos

Model	Average FPS in videos
Faster R-CNN	30

5.2 SSD MobileNet v2

The implementation of the SSD MobileNet v2 is done in Tensorflow. We trained this model for 60k steps with initial learning rate as 0.004 and batch size 24.

Table -3: Inference for SSD MobileNet v2 on images

Model	Inference Speed	COCO mAP
SSD MobileNet v2	10ms	22

Table -2: Inference for Faster R-CNN on videos

Model	Average FPS in videos
SSD MobileNet v2	75

6. INFERENCE

MobileNet v2 performed better in terms of speed on our machine. However, the accuracy of the Faster R-CNN model was better than the SSD MobileNet v2 model. In our results, in images with three explicit objects there are cases where the Faster R-CNN model correctly detects all three objects whereas the SSD MobileNet v2 model only correctly detects two objects. Tables 1 and 3 summarize our inferences on images. On videos, the Frames per second of SSD MobileNet v2 are considerably higher than the Faster R-CNN model, hence the SSD MobileNet v2 model can be used in real time for explicit content detection. Tables 2 and 4 summarize our inferences on videos.

7. CONCLUSION

This paper highlights the difference in performance between two models object detection models. Taking advantage of the considerably smaller size of MobileNet v2,

this model performed better as compared to Faster R-CNN in terms of speed especially on videos yielding more Frames per second on our test machine. Hence, MobileNet v2 can be used in real time object detection. However, with more training data, both models can perform considerably better.

REFERENCES

[1] Y LeCun. Y Bengio. and G Hinton. Deep learning. nat. 521, 436 – 444,2015.

[2] Geoffrey E Hinton. Alex Krizhevsky. and Ilya Sutskever. Imagenetclassification with deep convolutional neural networks. Advances in neural information processing systems, 25:1106–1114, 2012.

[3] Olga Russakovsky. Jia Deng. Hao Su. Jonathan Krause. SanjeevSatheesh. 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.

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

[5] Jonathan Long. Evan Shelhamer. and Trevor Darrell. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 3431–3440, 2015.

[6] Tsung-Yi Lin. Piotr Doll´ar. Ross Girshick. Kaiming He. Bharath Hariha-ran. and Serge Belongie. Feature pyramid networks for object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2117–2125, 2017.

[7] 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.

[8] Mark Sandler. Andrew Howard. Menglong Zhu. Andrey Zhmoginov.and Liang-Chieh Chen.Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE conference on computer visionand pattern recognition, pages 4510–4520, 2018.

[9] Ross Girshick. Fast r-cnn. In Proceedings of the IEEE international conference on computer vision, pages 1440–1448, 2015.

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

[11] Wei Liu. Dragomir Anguelov. Dumitru Erhan. Christian Szegedy. ScottReed. Cheng-Yang Fu. and Alexander C Berg. Ssd: Single shot multi boxdetector. In European conference on computer vision, pages 21–37.Springer, 2016.

[12] Joseph Redmon. Santosh Divvala. Ross Girshick. and Ali Farhadi. Youonly look once: Unified. real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages779–788, 2016.

[13] Andrew G Howard. Menglong Zhu. Bo Chen. Dmitry Kalenichenko.Weijun Wang. Tobias Weyand. Marco Andreetto. and Hartwig Adam.Mobilenets: Efficient convolutional neural networks for mobile vision applications.arXiv preprint arXiv:1704.04861, 2017.

[14] Ali Qamar Bhatti. Muhammad Umer. Syed Hasan Adil. MansoorEbrahim. Danyal Nawaz. and Faizan Ahmed. Explicit content detection system: An approach towards a safe and ethical environment. Applied Computational Intelligence and Soft Computing, 2018, 2018.

[15] Yu-Chun Lin. Hung-Wei Tseng. and Chiou-Shann Fuh. Pornography detection using support vector machine.In16th IPPR Conference on Computer Vision, Graphics and Image Processing (CVGIP 2003),volume 19, pages 123–130, 2003.

[16] Rigan Ap-Apid. An algorithm for nudity detection. In5th Philippine Computing Science Congress, pages 201–205, 2005.

[17] Hong Zhu. Shuming Zhou. Jianying Wang. and Zhongke Yin.Analgorithm of pornographic image detection. InFourth International Conference on Image and Graphics (ICIG 2007), pages 801–804. IEEE,2007.

[18] LabelImg Tzutalin. Git code (2015).

[19] Kaiming He. Xiangyu Zhang. Shaoqing Ren. and Jian Sun.Deepresidual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778,2016.

[20] Tsung-Yi Lin. Michael Maire. Serge Belongie. James Hays. PietroPerona. Deva Ramanan. Piotr Doll´ar. and C Lawrence Zitnick. Microsoftcoco: Common objects in context. InEuropean conference on computer vision, pages 740–755. Springer, 2014.

[21] Andrew G Howard. Menglong Zhu. Bo Chen. Dmitry Kalenichenko.Weijun Wang. Tobias Weyand. Marco Andreetto. and Hartwig Adam.Mobilenets: Efficient convolutional neural networks for mobile visionapplications.arXiv preprint arXiv:1704.04861, 2017.

[22] Christian Szegedy. Sergey Ioffe. Vincent Vanhoucke. and Alexander AAlemi.Inception-v4. inception-resnet and the impact of residual connections on learning. InThirty-first AAAI conference on artificial intelligence, 2017.