INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X

Impact factor: 4.295

(Volume 4, Issue 2)

Available online at: www.ijariit.com

Object recognition using CNN

Jagruti Jadhav
jagrutijadhav16@gmail.com
Padmabhushan Vasantdada Patil Pratishthan's College of Engineering, Mumbai, Maharashtra

Mehzabeen Attar
Mehzabeenattar@gmail.com
Padmabhushan Vasantdada Patil Pratishthan's College of Engineering, Mumbai, Maharashtra

Shradha Patil
shradhapatil2012@gmail.com
Padmabhushan Vasantdada Patil Pratishthan's College of Engineering, Mumbai, Maharashtra

Saleem Beg
msbsham95@gmail.com
Padmabhushan Vasantdada Patil Pratishthan's College of Engineering, Mumbai, Maharashtra

ABSTRACT

Object recognition is a popular task in computer vision. The method usually requires the presence of a data-set annotated with location information of the objects, which is in the form of bounding boxes around the objects. In this project, we have implemented a method to carry out object recognition in a weakly supervised manner i.e., using partially annotated data-set. The data-set provides the information about what objects are present in the image but not where they are present. We have used a Convolutional Neural Network (CNN) based architecture to perform this task. We also validated by experimenting with different architectures that mere information of presence/ absence of objects in an image (weak labels) does provide their location information for free.

Keywords: *Multi-object detection, Object recognition, Object recognition applications.*

1. INTRODUCTION

The modern world is enclosed with gigantic masses of digital visual information. To analyze and organize these devastating ocean of visual information image analysis techniques are major requisite. In particular useful would be methods that could automatically analyze the semantic contents of images or videos. The content of the image determines the significance in most of the potential uses. One important aspect of image content is the objects in the image. So there is a need for object recognition techniques.

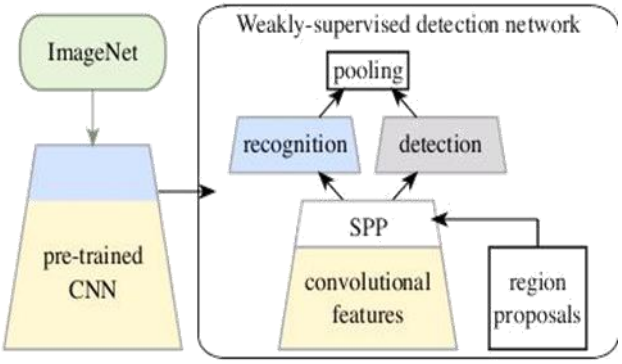
Object recognition is an important task in image processing and computer vision. It is concerned with determining the identity of an object being observed in an image from a set of known tags. Humans can recognize any object in the real world easily without any efforts; on contrary machines by itself cannot recognize objects. Algorithmic descriptions of recognition task are implemented on machines; which is an intricate task.

2. WEAKLY SUPERVISED OBJECT RECOGNITION

A. Method overview

This method is implemented as described in paper.It uses the concept of transfer learning in the implementation of the CNN architecture. In this form of learning, a pre- trained architecture is incorporated into the current model and training is done only on few layers in the current model. This saves huge amount of training time. We have used a pre-trained architecture trained on ImageNet data-set in a manner used in the papers. The ImageNet database, consists of tightly cropped images of single objects which enables the pre-trained

architecture to recognise individual objects. Two fully connected adaptation layers are added at the end of the pre-trained architecture, which adapts the new combined architecture to recognise individual objects in a cluttered image with multiple objects in it.



Thus object recognition techniques need to be developed which are less complex and efficient.n this paper we look at how the power of CNNs can be leveraged in Weakly Supervised Detection (WSD), which is the problem of learning object detectors using only image-level labels. The ability of learning from weak annotations is very important for two reasons: first, image understanding aims at learning an growing body of complex visual concepts (e.g hundred thousand object categories in ImageNet). Second, CNN training is data-hungry. Our method starts from a CNN pre-trained for image classification on a large dataset, e.g. ImageNet. It then modifies to reason efficiently about region branching off a recognition and a detection data streams. The resulting architecture can be fine-tuned on target dataset to achieve state-of-the-art weakly supervised object detection using only image-level annotations.

B. Implementation

The method of weakly supervised object recognition has been implemented on an Nvidia GPU with 2GB RAM. The system ran CentOS and Torch was used to implement the CNN architectures. Training has been done on the train dataset provided by Pascal VOC2012. Testing on the test dataset of Pascal VOC2012 could have been done by uploading the results on their server and getting accuracies after a week. In order to save time, testing in this project has been done on the test dataset provided by Pascal VOC2007. Memory constraints required us to cut down the pre-trained model network architecture from 7 convolutional layers to 5 convolutional layers.

The training data-set had 9232 images while the test data-set had 2308 images. There were twenty classes in the data-set, the true positive rates, the true negative rates and the precision values for each class is mentioned in the table I. All values are in percentage terms.

Class	TP Rate	TN Rate	Precision
aeroplane	60	98.67	66.13
bicycle	4.4	99.89	68.75
bird	12.8	99.74	75.51
boat	28.98	99.48	67.11
bottle	0.83	99.6	9.52
bus	20.77	98.51	34.86
car	62.19	94.64	68.28
cat	70.48	93.25	42.86
chair	37.06	95.05	48.1
cow	9.45	99.73	48
dining table	0.81	13.33	0.89
dog	7.39	99.07	43.24
horse	0.36	99.88	14.28
motorbike	1.72	13.79	0.89
person	81.45	76.36	71.67
potted plant	1.97	99.83	38.46
sheep	32.65	99.24	46.38
sofa	4.51	99.59	45.71
train	20.46	99.08	55.21
tv monitor	14.51	48.1	0.89

TABLE I: Results of object recognition

From the accuracies tableI, we observe that the scores are not at par with the paper on which this method was based on. This may be because of the removal of two layers from the pre-trained architecture. Nonetheless, we observe satisfactory accuracies throughout the classes. We also observe that the person class is being predicted with considerably good rates than any other classes. The prediction

capability of a class depends on the number of training images available for that class.A variation of the described architecture has also been implemented with only three convolutional layers in the pre-trained architecture and four adaptation layers. With this form of architecture we observed that the accuracies reduced. This tells us that higher the number of layers in the pre-trained architecture better is the object recognition capability.

C. Experiments

All the images were padded with zeros to bring them to adimension of 500 × 500 to suite the CNN model.

The target- images, which were formed for every data-set image, were of the same dimension which is 500 × 500. This resulted in the dimension of the target-vector to be of dimension 250000 × 1. Such a large dimensional vector could not fit into the CNN architecture because of memory constraints. Two alternative measurements have been taken to address this problem.

1) Inherent Scaling between the inputs and the targets: The initial target-image which was of 500 × 500 dimension was scaled down to 100 × 150 resulting in the target-vector to be of dimension 15000×1. Thus the model has to now account also for the scale difference between the inputs and the targets. Several other scaling down factors were also considered.

2) Inputs and Targets of similar dimension: In this case, the input and the outputs were brought to same dimension by scaling down the images in the target data-set. The training and testing images which were initially around 500 × 500 dimension were scaled down to 100 × 150 dimension. Suitable target dataset was prepared. Further, as it was observed that the CNN model used a sliding window of 224×224 dimension, the inputs and the targets were also brought to the same dimension, however, memory constraints prevented us from executing this case. When the data-sets were trained using the above variations random patterns were obtained on the desired target data-sets. These patterns were varying slightly for every image. The integral of the resulting target-images resulted in some random number which did not correspond with the number of objects actually present in the image. A possible transformation to these patters can be done for them to depict the object counts.

3. RELATED WORK

In this section, we will be talking about CNN related work in object detection and the trend towards smaller CNN models.

CNN Architectures

Convolutional Neural Network (CNN) usually stands for the spatial filters. These filters are used for extracting features from pictures. Some well-known filters are neural network which contains one or more convolutional neural layers. Each neural layer can be regarded as acombination of several Histogram of Oriented Gradients (HOG) and color histograms, etc. A typical input for an convolutional layer is a 3- dimensional grid. They are height (H), width (W) and channels (C). Here each channel represents a filter in the convolutional layer. The input of first layer usually has a shape of (H, W, 3), where 3 stands for the RGB channels for the raw pictures. CNN became popular in visual recognition field when it is introduced by LeCun et al. for handwritten zip code recognition [11] in the late 90s. In their work, they used (5, 5, C)-size filters.

Later work proved that smaller filters have multiple advantages, such as less parameters and reducing the size of network activations. In a VGG network proposed by Karen Simonyan et al., (3, 3, C)- size filters are extensively used, while the networks such as Network-in-Networ and GoogLeNet widely adopt(1, 1,C)-size filters, the possibly smallest filters and used for compressing volume of the networks. With the networks go deep, the filter size design gradually become a problem that almost all the CNN practitioners have to face.

Hence, several schemes for network modularization are proposed. Such modules usually include multiple convolutional layers with different filter sizes and these layers are combined together by stack or concatenation. In a GoogLeNet architecture, such as [18, 19], (1, 1, C)-size, (3, 3, C)-size and (5, 5, C)-size are usually combined together to form an”Inception” module and even with filter size of (1, 3, C) or (3, 1, C). In addition to modularizing the network, communication and connections across multiple layers also improve the performance of the network. This seems to be a similar idea with Long Short Term Memory (LSTM) or Gated Recurrent Unit (GRU) architecture in Recurrent Neural Network (RNN). Residual Network (ResNet) and Highway Network adopted such ideas to allow connections to skip multiple layers. These”bypass” connections can effectively send back the gradients through multiple layers without any blocking in a backward propagation pass when necessary.

CNN for Object Detection

With the advancement of accuracy in image classification,the research for object detection also developed in a fast speed. Before 2013, feature extraction techniques such as , which proposed an combined application of HoG and SVM can achieve a high accuracy on the PASCAL data-set . In 2013, a fundamental revolution occurred in this field, which was caused by the introduction of Regionbased Convolutional Neural Networks (R-CNN), proposed by Girshick and Ross. R-CNN firstly proposes possible regions for residing objects, then makes use of CNN to classify objects in these regions. However, these two independent operations require high computation and make it time-consuming. An modification of R-CNN is made by Girshick and Ross, which is called fast R- CNN. This

architecture integrate the two independent tasks into one multi-task loss function, which accelerates the computation of proposals and classification. Later, a more integrated version of R-CNN, namely the faster R-CNN was proposed by Ren et al., which achieves more than 10x faster than the original R-CNN. A receSnt proposal, R- FCN with a fully convolutional layer as the final parameterized layer further shortens the computation time used for region proposals. R-CNN can be regarded as a cornerstone for the development of CNN for object detection. A large amount of work is based on this architecture and achieves great accuracy. However, arecent work shows that CNN based object detection can be even faster. YOLO (You Only Look Once) is such an architecture integrating region proposition and object classification into one single stage, which significantly contributes to simplification of the pipeline of object detection, as well as reduction of the total computation time.

Toward Smaller Models

With CNN goes deeper, more parameters need to be stored, which makes the model larger and larger. Deeper CNN and larger modules usually achieve a higher accuracy, but people wonder whether a small model can reach a similar accuracy as a large model. In this sub-section, we talk about several popular model compression techniques aiming to reduce the size of CNN models. As we know, singular value decomposition (SVD) is widely used to reduce matrix dimensionality. It is also introduced to pre-trained CNN models to reduce model size. Another approach reported is Network Pruning, proposed by Han et al., which prunes the parameters below a certain threshold to construct a sparse CNN. Recently, Han etal. have further improved their approach and proposed a new approach, Deep Compression, together with their hardware design to accelerate the computation of CNN models. A recent research called Squeeze Net even reveals that a complex CNN model as AlexNet accuracy can be compressed to smaller than 0.5 Mbytes.
Here are two examples of model compression.The fa- mous ImageNet winner VGG-19 model stores more than 500 Mbytes parameters, which achieves a top-5 accuracy of about 87% on ImageNet, while the equally famous ImageNet winner GoogLeNet-v1 only contains about 50 Mbytes parameters, achieving the same accuracy as VGG- 19. The well-known AlexNet model with a size of more than 200 Mbytes parameters, achieves about 80% top-5 accuracy on ImageNet image classification challenge, while the SqueezeNet model with a much smaller size, about 4.8 Mbytes parameters, can also achieve that accuracy. We can anticipate that there is much room left for compressing these CNN models, to better fit them to portable devices.

Methods

In this section, the CNN model to detect objects and the implementation of for the same is discussed below.

CNN Model

The model has the benefit of small model size, good energy efficiency and good accuracy due to the fact that it's fully convolutional and only contains a single forward pass. The overview of this object detection model is as following in Figure 1. The CNN model we adopted is called SqueezeDet. The SqueezeDet model isa fully convolutional neural network for object detection. It's based on SqueezeNet architecture that extracts feature maps from image with CNN.Then another convolutional layer is used to find bounding box coordinates, confidence score and class probabilities. Finally, a multi-target loss is applied to compute final loss in training phase and a NMS filter is enforced to reduce the number overlapping bounding boxes and generate final detection in evaluation phase.

Data-set and Features

The data-set we use is The KITTI Vision Benchmark Suite, which is made for academic use in the area of autonomous driving. For our target, we use the object detection data-set, which contains 7481 training Data augmentation is implemented in the model training including image flipping, random cropping, batch normalization. Figure 5 is a typical scenario image in the dataset.

an, as the 3x3 convolutional layer take 9 times more parameters. And if 3x3 has to be used for the sake of activation area, we want to limit the input layer size as much as we can. With 9 layers' fire modules, 2 layers of polling and 1 layer of dropout, the feature map for each image can be obtained. After that, a 1x1 convolutional layer is used to extract bounding box coordinates, class scores and confidence score. For each activation in feature map, it will generate K bounding boxeswith 4K bounding box coordinates (x1, y1, x2, y2). Each loss calculation in training and for final detection in inference.

4. CONCLUSION

In this project, we trained a CNN object detection model at desktop platform and applied the trained model into a mobile platform. As a baseline, we have a running Android app that runs our CNN model trained by Tensorflow offline. The model size is 8 MegaBytes and the achieved testing accuracy is 76.7%.

The interface between Tensorflow and Android is still not perfect as the latency caused by interface is longer than the actual computation time in the graph. In addition, there is no documentation for the interface. Google announced that they plan to release the "Tensorflow Lite" for mobile platform, so we expect these issues to be significantly improved.

5. ACKNOWLEDGEMENT

We are grateful to the anonymous guides for their valuable support. Our sincere thanks to Project Guide Prof. Sanas sir, Project Coordinator Prof. Vijaya mam.

6. REFERENCES

[1] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, volume 1, pages 886–893. IEEE, 2005.
[2] E. L. Denton, W. Zaremba, J. Bruna, Y. LeCun,and R. Fergus. Exploiting linear structure within convolutional networks for efficient evaluation. In Advances in Neural Information Processing Systems, pages 1269–1277, 2014.
[3] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. International journal of computer vision, 88(2):303– 338, 2010.
[4] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? The kitti vision benchmark suite. In Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, pages 3354–3361. IEEE, 2012.
[5] R. Girshick. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, pages 1440–1448, 2015.
[6] S. Han, J. Pool, J. Tran, andW. Dally. Learning both weights and connections for efficient neural network. In Advances in Neural Information Processing Systems, pages 1135– 1143, 2015.
[7] A. Harp. Tensorflow android camera demo. https: //github.com/tensorflow/tensorflow/tree/master/tens orflow/examples/android, 2016.
[8] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 770–778, 2016.
[9] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size.arXiv preprint arXiv: 1602.07360, 2016.
[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pages 1097–1105, 2012.

