A Simplistic All Convolution Net for Efficient Real Time Object Recognition

A project report submitted for the partial fulfillment of the
Bachelor of Technology Degree
in
Computer Science & Engineering
under
Maulana Abul Kalam Azad University of Technology
by

Shuvozit Ghose
Roll No: 10400116085, Registration Number: 161040110162
Ushnik Ghosh
Roll No: 10400116040, Registration Number: 161040110207

Academic Session: 2016-2020

Under the Supervision of
Prof. Shreejita Mukherjee

Department of Computer Science and Engineering
Institute of Engineering & Management
Y-12, Salt Lake, Sector 5, Kolkata, Pin 700091, West Bengal, India

Affiliated To

Maulana Abul Kalam Azad University of Technology
BF 142, BF Block, Sector 1, Kolkata, West Bengal 700064

May 2020

INSTITUTE
OF ENGINEERING & MANAGEMENT
श्रद्धावान लभते ज्ञानम्
Salt Lake Electronics Complex, Kolkata - 700091, WB, INDIA

Phone : (033) 2357 2969/2059/2995
(033) 2357 8189/8908/5389
Fax : 91 33 2357 8302
E-mail : director@iemcal.com
Website : www.iemcal.com

CERTIFICATE

TO WHOM IT MAY CONCERN

This is to certify that the project report titled **"A Simplistic All Convolution Net For Efficient Real Time Object Recognition"**, submitted by **Shuvozit Ghose, Roll No: 10400116085, Registration Number: 161040110162**, **Ushnik Ghosh, Roll No: 10400116040, Registration Number: 161040110207**, students of **Institute of Engineering & Management** in partial fulfillment of requirements for the award of the degree of **Bachelor of Technology** in **Computer Science & Engineering**, is a bona fide work carried out under the supervision of **Prof. Shreejita Mukherjee** during the final year of the academic session of 2016-2020. The content of this report has not been submitted to any other university or institute for the award of any other degree.

It is further certified that the work is entirely original and the performance has been found to be satisfactory.

Prof. Shreejita Mukherjee
Assistant Professor
Department of Computer Science and Engineering
Institute of Engineering & Management

Prof. (Dr.) Sourav Saha
H.O.D.
Department of Computer Science and Engineering
Institute of Engineering & Management

Prof.(Dr.) Amlan Kusum Nayak
Principal
Institute of Engineering & Management

INSTITUTE OF ENGINEERING & MANAGEMENT

DECLARATION
FOR NON-COMMITMENT OF PLAGIARISM

We, **Shuvozit Ghose**, **Ushnik Ghosh**, students of B.Tech. in the Department of Computer Science and Engineering, Institute of Engineering & Management have submitted the project report in partial fulfillment of the requirements to obtain the above noted degree. We declare that we have not committed plagiarism in any form or violated copyright while writing the report and have acknowledged the sources and/or the credit of other authors wherever applicable. If subsequently it is found that we have committed plagiarism or violated copyright, then the authority has full right to cancel/reject/revoke our degree.

Name of the Student: SHUVOZIT GHOSE

Full Signature: shuvozit ghose

Name of the Student: USHNIK GHOSH

Full Signature: Ushnik Ghosh

Date: 08-07-2020

Abstract

Object detection is a very widely known problem in the context of the pattern recognition and computer vision because of its immense applications in real-world. The Convolution Neural Networks (CNN) are the most popular framework for the task of object recognition as it has superior capability of visual feature extraction. However, most modern convolutional neural networks (CNNs) used for object recognition are built using the same principles: Alternating convolution and max-pooling layers followed by a small number of fully connected layers. We re-evaluate the state of the art for object recognition from small images with convolutional networks, questioning the necessity of different components in the pipeline. We find that max-pooling can simply be replaced by a convolutional layer with increased stride without loss in accuracy on several image recognition benchmarks. Following this finding and building on other recent work for finding simple network structures, we propose a new architecture that consists solely of convolutional layers and yields competitive or state of the art performance on several object recognition datasets (CIFAR-10, CIFAR-100, ImageNet). To analyze the network, we introduce a new variant of the "deconvolution approach" for visualizing features learned by CNNs, which can be applied to a broader range of network structures than existing approaches.

Acknowledgements

We must not forget to acknowledge everyone who has provided constant support to us during our B.Tech course. First and foremost, we would like to express sincere gratitude to our supervisor **Prof. Shreejita Mukherjee** for his continuous support and motivation in fueling the pursuance of carrying out this project endeavor. Without his guidance and persistent encouragement, this project work would not have been possible. He has been a tremendous mentor for us throughout this academic journey. Many of his academic advises about our career growth have been priceless.

We would like to convey sincere gratitude to **Prof. Sourav Saha** for providing us constant inspiration to stand firm against several setbacks throughout the course. Additionally, we would like to thank all the technical, non-technical and office staffs of our department for extending facilitating cooperation wherever required. We also express gratitude to all of our friends in the department for providing the friendly environment to work on the project work.

We would also like to thank our Director **Prof. Satyajit Chakraborti** for providing us an outstanding platform in order to develop our academic career. In addition, we also preserve a very special thankful feeling about our Principal **Prof. Amlan Kusum Nayak** for being a constant source of inspiration.

A special thank is due to our family. Words cannot express how grateful we are to our parents for all the sacrifices that they have made while giving us necessary strength to stand on our own feet.

Finally, we would like to thank everybody who has provided assistance, in whatever little form, towards successful realization of this project but with an apology that we could not mention everybody's name individually.

Contents

List of Figures

List of Tables

1

Introduction

1.1 Motivation

Object detection is a very widely known problem in the context of the pattern recognition and computer vision. It is basically a computer vision technique for locating instances of objects in images or videos. This technology has the power to classify just one or several objects within a digital image at once. Object detection has been around for years, but is becoming more apparent across a range of industries now more than ever before. When humans look at images or video, we can recognize and locate objects of interest within a matter of moments. The goal of object detection is to replicate this intelligence using a computer. Object Detection recognizes instances of a predefined set of object classes by using bounding boxes. Compared to Image Classification, Object Detection is considerably more complicated due to the simple fact that an image can have anywhere from zero to dozens of objects in them. Which in turn, means that during training, an Object Detection model can output more than one prediction for a single image. Object detection is a key technology behind advanced driver assistance systems (ADAS) that enable cars to detect driving lanes or perform pedestrian detection to improve road safety. Object detection is also useful in applications such as video surveillance or image retrieval

systems. Object detection is breaking into a wide range of industries, with use cases ranging from personal security to productivity in the workplace. Facial detection is one form of it, which can be utilized as a security measure to let only certain people into a highly classified area of a government building. It can be used to count the number of people present within a business meeting to automatically adjust other technical tools that will help streamline the time dedicated to that particular meeting. It can also be used within a visual search engine to help consumers find a specific item they're on the hunt for – Pinterest is one example of this, as the entire social and shopping platform is built around this technology. We recently debuted Intelligent Vision 2.0, a suite of AI products that improve collaboration by delivering accurate info to help further automate workflows. These features utilize people and object detection to create big data for a variety of applications in the workplace. At present, Convolution neural based frameworks are the most efficient model for object detection. But these models are not suitable for real time object detection as it involves process heavy calculations. Actually, the pooling layers are responsible for this kind of complexions. To alleviate this problem, we have proposed a simple all CNN framework for object recognition and shown that max-pooling can simply be replaced by a convolutional layer with increased stride without loss in accuracy on several image recognition benchmarks.

1.2 Related Work

Traditional Methods: To recognize different objects, we need to extract visual features which can provide a semantic and robust representation. SIFT, HOG and HOG-like features are the representative ones. This is due to the fact that these features can produce representations associated with complex cells in human brain. However, due to the diversity of appearances, illumination conditions and backgrounds, it's difficult to manually design a robust feature descriptor to perfectly describe all kinds of objects. Besides, a classifier is needed to distinguish a target object from all the other categories and to make the representations more hierarchical, semantic and informative for visual

recognition. Usually, the Supported Vector Machine (SVM), AdaBoost and Deformable Part-based Model (DPM) are good choices. Among these classifiers, the DPM is a flexible model by combining object parts with deformation cost to handle severe deformations. In DPM, with the aid of a graphical model, carefully designed low-level features and kinematically inspired part decompositions are combined. And discriminative learning of graphical models allows for building high-precision part-based models for a variety of object classes. Based on these discriminant local feature descriptors and shallow learnable architectures, state of the art results has been obtained on PASCAL VOC object detection competition and real-time embedded systems have been obtained with a low burden on hardware. However, small gains are obtained during 2010-2012 by only building ensemble systems and employing minor variants of successful methods.

Deep Learning Methods: Generic object detection aims at locating and classifying existing objects in any one image, and labelling them with rectangular bounding boxes to show the confidences of existence. The frameworks of generic object detection methods can mainly be categorized into two types. One follows traditional object detection pipeline, generating region proposals at first and then classifying each proposal into different object categories. The other regards object detection as a regression or classification problem, adopting a unified framework to achieve final results (categories and locations) directly. The region proposal-based methods mainly include R-CNN, SPP-net, Fast R-CNN, Faster R-CNN, R-FCN, FPN and Mask R-CNN, some of which are correlated with each other (e.g. SPP-net modifies RCNN with a SPP layer). The regression classification-based methods mainly include MultiBox, AttentionNet, G-CNN, YOLO, SSD, YOLOv2, DSSD and DSOD. The correlations between these two pipelines are bridged by the anchors introduced in Faster RCNN. The region proposal-based framework, a two-step process, matches the attentional mechanism of human brain to some extent, which gives a coarse scan of the whole scenario firstly and then focuses on regions of interest. Among the pre-related works, the most representative one is Over feat. This model inserts CNN into sliding window method, which predicts bounding boxes

directly from locations of the topmost feature map after obtaining the confidences of underlying object categories. The R-CNN adopts selective search to generate about 2k region proposals for each image. The selective search method relies on simple bottom-up grouping and saliency cues to provide more accurate candidate boxes of arbitrary sizes quickly and to reduce the searching space in object detection.

Popular Object Recognition Frameworks: The vast majority of modern convolutional neural networks (CNNs) used for object recognition are built using the same principles: They use alternating convolution and max-pooling layers followed by a small number of fully connected layers [1, 2,3]. Within each of these layers piecewise-linear activation functions are used. The networks are typically parameterized to be large and regularized during training using dropout. A considerable amount of research has over the last years focused on improving the performance of this basic pipeline. Among these two major directions can be identified. First, a plethora of extensions were recently proposed to enhance networks which follow this basic scheme. Among these the most notable directions are work on using more complex activation functions [4, 5, 6] techniques for improving class inference [7, 8] as well as procedures for improved regularization [9, 10, 11] and layer-wise pre-training using label information [12]. Second, the success of CNNs for large scale object recognition in the ImageNet challenge [13] has stimulated research towards experimenting with the different architectural choices in CNNs. Most notably the top entries in the 2014 ImageNet challenge deviated from the standard design principles by either introducing multiple convolutions in between pooling layers [14] or by building heterogeneous modules performing convolutions and pooling at multiple scales in each layer [15]. Since all of these extensions and different architectures come with their own parameters and training procedures the question arises which components of CNNs are actually necessary for achieving state of the art performance on current object recognition datasets. We take a first step towards answering this question by studying the simplest architecture we could conceive: a homogeneous network solely consisting of convolutional layers, with occasional dimensionality reduction by using a stride of 2.

1.3 Objective of the Proposed Work

Most modern convolutional neural networks (CNNs) used for object recognition are built using the same principles: Alternating convolution and max-pooling layers followed by a small number of fully connected layers. Surprisingly, we find that this basic architecture – trained using vanilla stochastic gradient descent with momentum – reaches state of the art performance without the need for complicated activation functions, any response normalization or max-pooling. That why we re-evaluate the state of the art for object recognition from small images with convolutional networks, questioning the necessity of different components in the pipeline. We find that max-pooling can simply be replaced by a convolutional layer with increased stride without loss in accuracy on several image recognition benchmarks. Following this finding – and building on other recent work for finding simple network structures-we propose a new architecture that consists solely of convolutional layers and yields competitive or state of the art performance on several object recognition datasets (CIFAR-10, CIFAR-100, ImageNet). To analyze the network, we introduce a new variant of the "deconvolution approach" for visualizing features learned by CNNs, which can be applied to a broader range of network structures than existing approaches. We empirically study the effect of transitioning from a more standard architecture to our simplified CNN by performing an ablation study on CIFAR-10. Our results both confirm the effectiveness of using small convolutional layers as recently proposed [14] and give rise to interesting new questions about the necessity of pooling in CNNs. Since dimensionality reduction is performed via strided convolution rather than max-pooling in our architecture it also naturally lends itself to studying questions about the invertibility of neural networks [16]. For a first step in that direction we study properties of our network using a deconvolutional approach similar to [9].

1.4 Organization of the Project Report

CHAPTER 1. INTRODUCTION: Provides a brief introduction on what is being done and why this topic has been chosen. We have described the significance of Object Recognition as well as it's applications. This section also focuses on our objective and the organization of our project. In this section we shall also discuss a literature survey of related works done on similar topics and we have analyzed some related works consisting of traditional and deep learning methods respectively along with along with popular object recognition framework.

CHAPTER 2. PROPOSED FRAMEWORK: In this section, the methods of how to go about the entire technique is described. In this section we begin with analyzing most modern convolutional neural networks (CNNs) used for object recognition built using the principle "Alternating convolution and max-pooling layers followed by a small number of fully connected layers". We re-evaluate the state of the art for object recognition from small images with convolutional networks, questioning the necessity of different components in the pipeline. To analyses the network, we introduce a new variant of the "deconvolution approach" for visualizing features learned by CNNs, which can be applied to a broader range of network structures than existing approaches. We discuss various All CNN Models and compare their network architecture. We also introduce All CNN for both single class classification and multi class classification.

CHAPTER 3. EXPERIMENTAL RESULTS AND ANALYSIS: This section produces a report of how our framework is performing. In order to evaluate the performance of the proposed shape retrieval framework, experiments have been conducted based on CIFAR-10 dataset. We have used accuracy and error percentage as our performance evaluation metric.

CHAPTER 4. CONCLUSION: After obtaining the results, this is the section where we draw conclusions and discuss the limitations of our framework and future scope of the work i.e. where and how we can improve this technique to make it more suitable.

2

Proposed Framework

2.1 Overview of Proposed Framework

This section discusses our proposed framework in detail. The proposed model for object recognition initially extracts the visual features of an image object using deep convolution filter banks and outputs high dimensional feature vector. These feature vectors are later used by the classifier for the object detection purpose. The advantage of our model is that we have discarded the max pooling layer. The description of each step of the overall flow is detailed next.

2.2 Convolution Neural networks

Convolutional Neural Networks are very similar to ordinary Neural Networks: they are made up of neurons that have learnable weights and biases. Each neuron receives some inputs, performs a dot product and optionally follows it with a non-linearity. The whole network still expresses a single differentiable score function: from the raw image pixels on one end to class scores at the other. And they still have a loss function (e.g. SVM/Softmax) on the last (fully-connected) layer and all the tips/tricks we developed for learning regular Neural Networks still apply. The Convolution layer accepts a input image of size W1×H1×D1 where W1, H1 and D1 are the width, height and depth respectively.

The layer requires two hyperparameters: they are Number of filters K, their spatial extent F, the stride S and the amount of zero padding P respectively. It produces a volume size W2×H2×D2 where:

$$W_2 = \frac{W1-F+2P}{S+1}$$

$$H_2 = \frac{H1-F+2P}{S+1}$$

Here, D1=K and W2, H2 and D2 are the width, height and depth of the output image respectively. In the output volume. The d-th depth slice (of size W2×H2) is the result of performing a valid convolution of the d-th filter over the input volume with a stride of S, then offset by d-th bias.

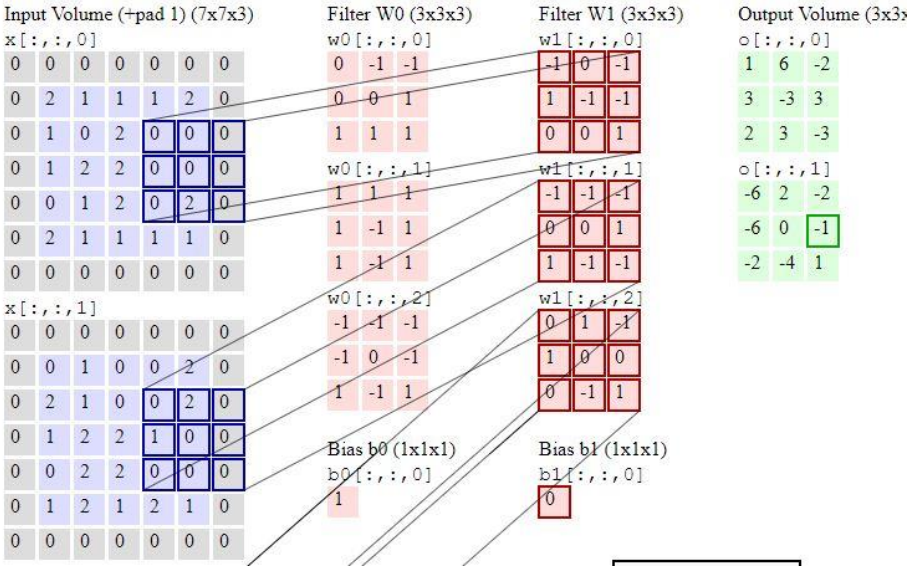

Fig 2.1: In depth feature extraction operation of the CNN layer

When dealing with high-dimensional inputs such as images, it is impractical to connect neurons to all neurons in the previous volume. Instead, we will connect each neuron to only a local region of the input volume. The spatial extent of this connectivity is a hyperparameter called the receptive field of the neuron (equivalently this is the filter size). The extent of the connectivity along the depth axis is always equal to the depth of the

input volume. It is important to emphasize again this asymmetry in how we treat the spatial dimensions (width and height) and the depth dimension: The connections are local in space (along width and height), but always full along the entire depth of the input volume.

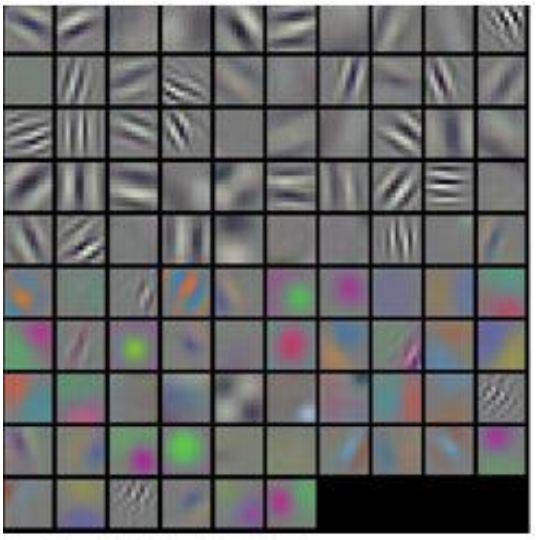

Fig 2.2: The visualization of the CNN filter banks

The main target of training is to extract the meaningful information of the context in the purpose of object recognition as shown in the fig 2.2.

2.3 Max Pooling

It is common to periodically insert a Pooling layer in-between successive Convolution layers in a Convolution Neural Networks architecture. Its function is to progressively reduce the spatial size of the representation to reduce the number of parameters and computation in the network, and hence to also control overfitting. The Pooling Layer operates independently on every depth slice of the input and resizes it spatially, using the MAX operation. The most common form is a pooling layer with filters of size 2x2 applied with a stride of 2 down samples every depth slice in the input by 2 along both width and height, discarding 75% of the activations. Every MAX operation would in this case be taking a max over 4 numbers (little

2x2 region in some depth slice). The depth dimension remains unchanged. The pooling layer accepts a input image of size W1×H1×D1 where W1, H1 and D1 are the width, height and depth respectively. The two hyperparameters: spatial extent F and,the stride S are used to produce a output image size W2×H2×D2 where:

$$W_2 = \frac{(W1-F)}{S+1}$$

$$H_2 = \frac{H1-F}{S+1}$$

Here, D1=D2 and W2, H2 and D2 are the width, height and depth of the output image respectively. In the output volume. The max pooling opration introduces zero parameters since it computes a fixed function of the input. For Pooling layers, it is not common to pad the input using zero padding. The max pooling operation normally reduces the dimensionality by discarding the irrelevant context information as shown in the fig 2.3.

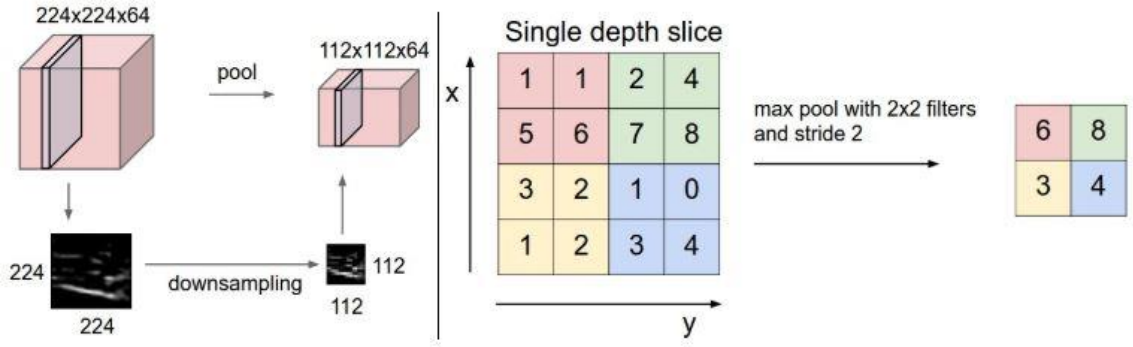

Fig 2.3: Reduction of dimensionality by Max Pooling operation

2.4 Deconvolution Approach

In order to analyze the network that we trained on ImageNet – and get a first impression of how well the model without pooling lends itself to approximate inversion – we use a 'deconvolution' approach. We start from the idea of using a deconvolutional network for visualizing the parts of an image that are most discriminative for a given unit in a network, an approach recently proposed by [14]. Following this initial attempt – and observing that

it does not always work well without max-pooling layers – we propose a new and efficient way of visualizing the concepts learned by higher network layers. The deconvolutional network (’deconvnet’) approach to visualizing concepts learned by neurons in higher layers of a CNN can be summarized as follows. Given a high-level feature map, the ’deconvnet’ inverts the data flow of a CNN, going from neuron activations in the given layer down to an image.

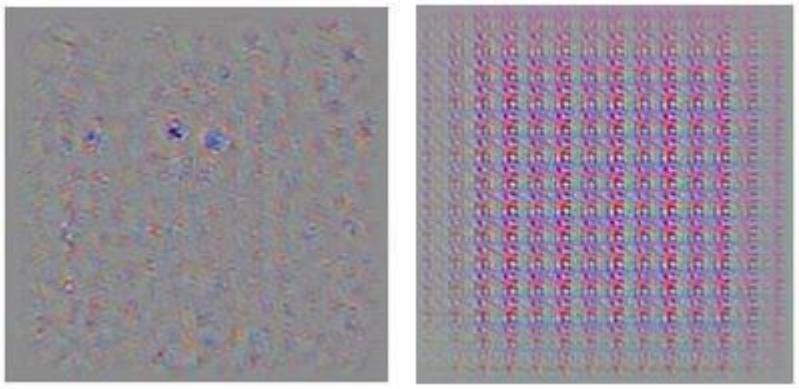

Fig 2.4: left image describes the features of deconvolution with pooling layer and right image describes the features of deconvolution without pooling layer

Typically, a single neuron is left non-zero in the high-level feature map. Then the resulting reconstructed image shows the part of the input image that is most strongly activating this neuron (and hence the part that is most discriminative to it). In order to perform the reconstruction through max-pooling layers, which are in general not invertible, the method of [14] requires first to perform a forward pass of the network to compute ’switches’ – positions of maxima within each pooling region. These switches are then used in the ’deconvnet’ to obtain a discriminative reconstruction. By using the switches from a forward pass, the ‘deconvnet’ is hence conditioned on an image and does not directly visualize learned features. Our architecture does not include max-pooling, meaning that in theory we can deconvolve without switches, i.e. not conditioning on an input image. This way we get insight into what lower layers of the network learn. The deconvolutional visual feature of the pooling layer is shown in the fig 2.4.

2.5 All CNN Networks

The All CNN networks basically composed of all the CNN layers by discarding the max pooling layers. Below we have discussed the various All CNN Models in the context of objection recognition. We also discuss the All CNN models for Single Class Classification and Multi-Class Classification.

2.5.1 Various All CNN Models

We have proposed three model for the purpose of object detection as shown in the table 1. The **A** model uses 5 x 5 convolution feature extractor layer. On the other hand, C model uses 5 x 5 convolution feature extractor layer. The B model uses 5 x 5 convolution feature extractor layer with 1 x 1 convolution feature extractor layer. Some people are at first confused to see 1x1 convolutions especially when they come from signal processing background. Normally signals are 2-dimensional so 1x1 convolutions do not make sense (it's just pointwise scaling). However, in ConvNets this is not the case because one must remember that we operate over 3-dimensional volumes, and that the filters always extend through the full depth of the input volume. For example, if the input is [32x32x3] then doing 1x1 convolutions would effectively be doing 3-dimensional dot products (since the input depth is 3 channels).

A	B	C
	Input 32 × 32 RGB image	
5 × 5 conv. 96 ReLU	**5 × 5 conv. 96 ReLU** **1 × 1 conv. 96 ReLU**	**3 × 3 conv. 96 ReLU** **3 × 3 conv. 96 ReLU**
	3 × 3 max-pooling stride 2	
	5 × 5 conv. 192 ReLU	
5 × 5 conv. 192 ReLU	**5 ×5 conv. 192 ReLU** **1 × 1 conv. 192 ReLU**	**3 × 3 conv. 192 ReLU** **3 ×3 conv. 192 ReLU**
	3 ×3 max-pooling stride 2	
	3 × 3 conv. 192 ReLU	
	1 ×1 conv. 192 ReLU	

	1 ×1 conv. 10 ReLU	
	global averaging over 6×6 spatial dimensions	
	10 or 100-way softmax	

Table 2.1: Various All CNN models

2.5.2 All CNN for Single Class Classification

For the single class classification of our network, we have used CIFAR-10 dataset which consists of 60000 32x32 color images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images. The dataset is divided into five training batches and one test batch, each with 10000 images. The test batch contains exactly 1000 randomly-selected images from each class. The training batches contain the remaining images in random order, but some training batches may contain more images from one class than another. Between them, the training batches contain exactly 5000 images from each class. The classes in the dataset, as well as 10 random images from each class is shown in figure 2.5.

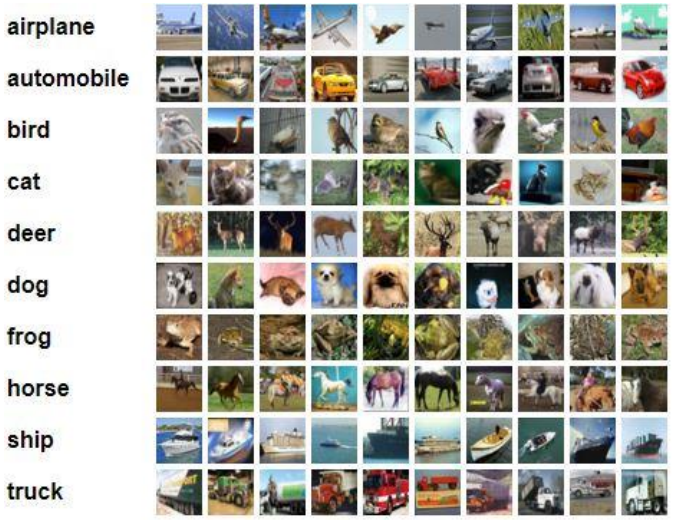

Fig 2.5: Classes of CIFAR-10 dataset

2.5.3 All CNN for Multi-Class Classification

As our All CNN model is very effective for the small images singular class classification. We can perform multi-class classification by adding three pre-processing steps in the

existing model. In the first step, we have to add Region Proposal Network that will generate the patches of the object context region. In the second step, we have to reduce the dimensionality of every patch. In the final step, we have to feed the patches in the All CNN Network and map the outputs sequentially.

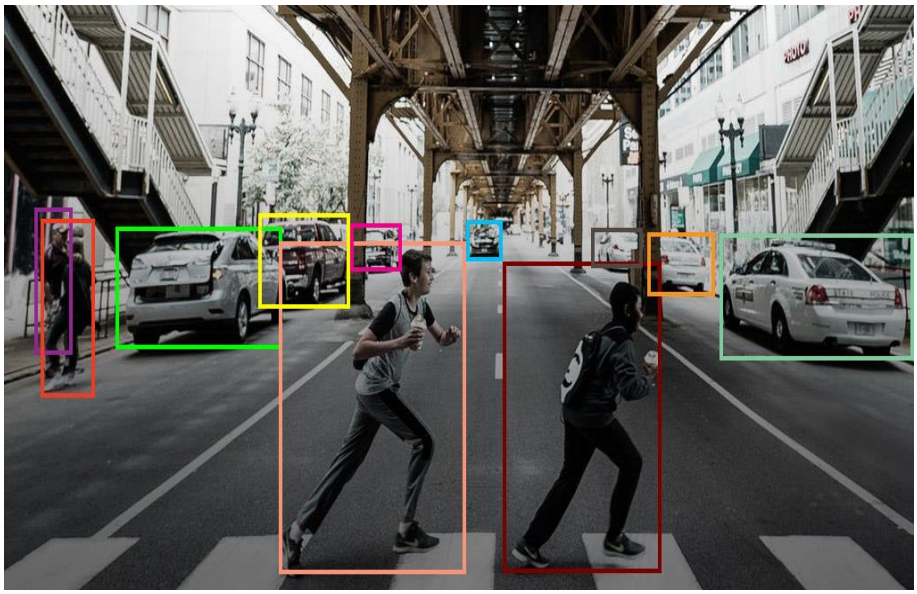

Fig 2.6: Example of the Multi-class classification image

3

Experimental Results and Discussion

3.1 Experimental Setup

In order to evaluate the performance of the proposed shape retrieval framework, experiments have been conducted based on CIFAR-10 dataset. The dataset comprises 60000 images in 10 classes, with 6000 images per class and there are 50000 training images and 10000 test images. In order to quantify the effect of simplifying the model architecture we perform experiments on CIFAR-10 dataset. we use CIFAR-10 to perform an in-depth study of different models, since a large model on this dataset can be trained with moderate computing costs. We then test the best model found on CIFAR-10 with and without augmentations. We performed all experiments using the TensorFlow framework. The input image size of our network is 32 x 32. We have used stochastic gradient descent (SGD) optimizer with learning rate 0.01, momentum 0.9, decay rate of 0.0001 while training our model in an end-to-end fashion.

3.2 Performance Evaluation Metric

We have used accuracy and error percentage as our performance evaluation metric. The loss is calculated using binary cross entropy loss function. The mathematical definition of the error percentage, accuracy and binary cross entropy is given below:

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total number of predictions made}}$$

$$\text{Error percentage} = \frac{\text{Number of False Predictions}}{\text{Total number of predictions made}}$$

$$\text{loss} = -\frac{1}{N}\sum_{i=1}^{N} \log(p(y_i)) + (1 - y_i)\log(1 - p(y_i))$$

We Here, y_i is the output of our model and loss is the binary cross entropy loss.

3.3 Experimental Results

In experiments on CIFAR-10, we use three different base network models which are intended to reflect current best practices for setting up CNNs for object recognition. Architectures of these networks are described in section 2.5.1. Starting from model A

Model	Accuracy	Error%	Parameter
A	100-error	10.30%	≈1.28 M
B	100-error	9.10%	≈1.35 M
C	100-error	9.08%	≈1.4 M

Table 3.1: Experimental Results

(the simplest model) the depth and number of parameters in the network gradually increases to model C. Several things are to be noted here. First, as described in the table, all base networks we consider use a 1-by-1 convolution at the top to produce 10 outputs of which we then compute an average over all positions and a SoftMax to produce class-probabilities. We performed additional experiments with fully connected layers instead of 1-by-1 convolutions but found these models to consistently perform 0.5% - 1% worse than their fully convolutional counterparts. it can be observed that model B only one 1-by-1 convolution is performed after each “normal” convolution layer. Third, model C replaces all 5 x 5 convolutions by simple 3 x 3 convolutions. This serves two purposes: 1) it unifies the architecture to consist only of layers operating on 3 x 3 spatial neighborhoods of the previous layer feature map (with occasional subsampling); 2) if max-pooling is replaced

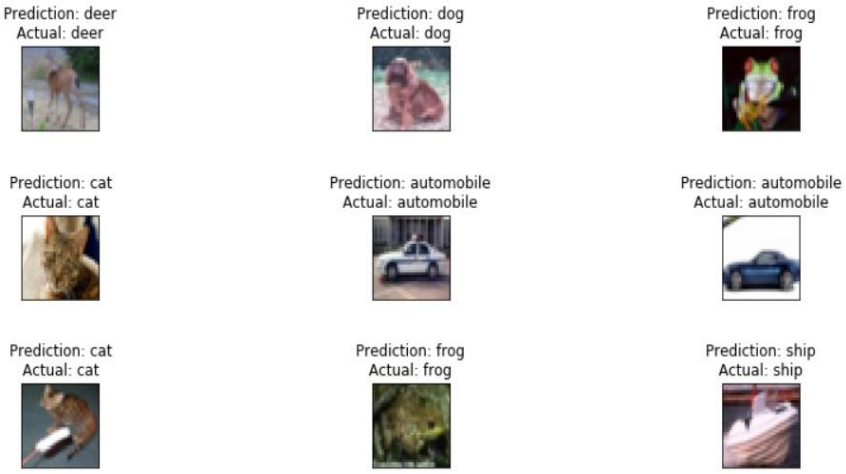

Fig 3.1: Output of our model

by a convolutional layer, then 3 x 3 is the minimum filter size to allow overlapping convolution with stride 2. We also highlight that model C resembles the very deep models. The experimental result is shown in the Table 3.1. The visual performance is shown in the Fig 3.1.

4

Conclusion

All CNN Network is proposed for real time object detection in this report. Our network is a simple model composed of all the CNN layers. The main advantage of our model is that it does not use the max pooling layer which is computational heavy. The proposed framework is simple and easy to implement. It is capable of detecting objects in the real-world scenario. We demonstrate the effectiveness of our system by conducting experiments on publicly available CIFAR-10 datasets. The main limitation of our model is that it is only capable of recognizing the low-resolution object image because of the training in the small low-resolution dataset of CIFAR-10. Although we have done experiment on small dataset due to the computation constrain, we have further plan to perform experiment on ImageNet dataset.

Bibliography

1) Jarrett, Kevin, et al. "What is the best multi-stage architecture for object recognition?." 2009 IEEE 12th international conference on computer vision. IEEE, 2009.
2) Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems. 2012.
3) Cireşan, Dan C., et al. "High-performance neural networks for visual object classification." arXiv preprint arXiv:1102.0183 (2011).
4) Goodfellow, Ian J., et al. "Maxout networks." arXiv preprint arXiv:1302.4389 (2013).
5) Lin, Min, Qiang Chen, and Shuicheng Yan. "Network in network." arXiv preprint arXiv:1312.4400 (2013).
6) Srivastava, Rupesh K., et al. "Compete to compute." Advances in neural information processing systems. 2013.
7) Stollenga, Marijn F., et al. "Deep networks with internal selective attention through feedback connections." Advances in neural information processing systems. 2014.
8) Srivastava, Nitish, and Ruslan R. Salakhutdinov. "Discriminative transfer learning with tree-based priors." Advances in neural information processing systems. 2013.
9) Zeiler, Matthew D., and Rob Fergus. "Stochastic pooling for regularization of deep convolutional neural networks." arXiv preprint arXiv:1301.3557 (2013).
10) Springenberg, Jost Tobias, and Martin Riedmiller. "Improving deep neural networks with probabilistic maxout units." arXiv preprint arXiv:1312.6116 (2013).
11) Wan, Li, et al. "Regularization of neural networks using dropconnect." International conference on machine learning. 2013.
12) Lee, Chen-Yu, et al. "Deeply-supervised nets." Artificial intelligence and statistics. 2015.
13) Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems. 2012.
14) Simonyan, Karen, Andrea Vedaldi, and Andrew Zisserman. "Deep inside convolutional networks: Visualising image classification models and saliency maps." arXiv preprint arXiv:1312.6034 (2013).
15) Szegedy, Christian, et al. "Going deeper with convolutions." Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.
16) Bruna, Joan, Arthur Szlam, and Yann LeCun. "Signal recovery from pooling representations." arXiv preprint arXiv:1311.4025 (2013).