Video Event Localisation and Classification

A Project Report

submitted by

G K SUDHARSHAN

in partial fulfilment of the requirements
for the award of the degree of

MASTER OF TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY, MADRAS.

May 2015

THESIS CERTIFICATE

This is to certify that the thesis entitled **Video Event Localisation and Classification**, submitted by **G K Sudharshan**, to the Indian Institute of Technology, Madras, for the award of the degree of **Master of Technology**, is a bonafide record of the research work carried out by him under my supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Prof. Hema A Murthy
Research Guide
Professor
Dept. of Computer Science and Engineering
IIT-Madras, 600 036

Place: Chennai

Date:

ACKNOWLEDGEMENTS

I would like to thank my co-partner Abil N George for his support and contribution in the toolkit development. Also like to thank all colleagues in DONLAb(IIT Madras) who helped me throughout my research. Lastly, thanks to my parents for all the moral support and the amazing chances they've given me over the years.

ABSTRACT

KEYWORDS: Convolutional Neural Network, Spatio Temporal Volumne, Saliency Estimation,

In this thesis we focus on detecting and identifying the multiple events occurring at a given instance. Discussion about the approaches followed to localize the multiple events occuring at given instance based on the saliency and motion based information. Spatio temporal volumes are extracted from a video segments where each volume corresponding to a specific event would be trained through 3D convolutional neural network model (3D-CNN) for sake of classification.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

CNN	Convolutional Neural Network
STV	Spatio Temporal Volume

CHAPTER 1

INTRODUCTION

In the era of Google glass, people want everything in front of them to be explicable. In domains like surveillance, detecting multiple events at same instance is helpful, for instance system must be capable to capture the fire accident and the person responsible for it simultaneously and then alert accordingly. Similarly, annotating events also helps in an effective video retrieval system to associate high level information in video along with the textual description.

In our problem definition, we are given Internet videos labeled with an event class, where the label specifies the events that occurs within video. In most of the dataset these are weakly labeled settings, i.e we do not have spatio-temporal segmentation, indicating coordinates and time points where at which event occurs. The detection aspect of our problem manifests task of localizing the event within the video further building the spatio-temporal volume for better event prediction.

In section II, we we will discuss about the existing techniques used for event detection. We would also cover about our implementation of convolutional neural network in section III. In section IV, we will describe the steps for extracting spatio-temporal volume.

CHAPTER 2

CONVOLUTIONAL NEURAL NETWORK

In this section we will discuss about the reasons for chosing the convolutional neural network and discuss the features supported by the indigenous neural network that was designed.

2.1 Why CNN ?

Covolutional neural networks (CNN) are variant of multiple layer perceptron that are designed by studying the complex arrangement of cells in the cats visual cortex. It fits very well onto the visual recognition domain where we expect the model to handle very high dimensional data, exploit the topology of image or video and be invariant to small translation and illumination changes. CNN leverage following concepts,

Local Connectivity

It exploits the spatially local correlation by enforcing local connectivity pattern between neurons of adjacent layers. In other words, every hidden units is only sensitive to a small block in the visual field, called receptive field. This drastically reduces the number of connections between input and hidden layer, following which diminishes the number of parameters needed to train the model.

Shared Filters

The hidden units are associated to the receptive field by filters which are shared within a feature map. These filters tries to capture edge like patterns within the receptive field. Additionally, sharing filters increases learning efficiency by greatly reducing the number of free parameters to be learned. Apart from reducing parameters, they extract the same feature at every position, which makes every feature map to be equi-variant to any changes in the input. The shared filters are associated to the receptive field by a dot product operation which can be expressed as a discrete convolution operation.

Pooling/Sub-sampling Hidden Units

According to this concept, we try to pool the hidden in non-overlapping neighborhood. Among the techniques average and max pooling, max pooling has been commonly used as it provides local translation in-variance. Pooling also reduces inputs to next layer of feature extraction, thus allowing us to have many more feature maps. All feature maps in latter layer extracts coarser features.

All these concepts enable CNN to achieve better generalization in the vision problems. We stack multiple such layers to achieve better responsiveness to larger visual field.

2.2 Python-DNN Toolkit

All these concepts enable CNN to achieve better generalization in the vision problems. We stack multiple such layers to achieve better responsiveness to larger visual field.

2.2.1 Implementation

Implementation of CNN is done in python using numerical computation library named *Theano*. It provides platform to run efficiently in CPU and GPU architecture. Following are some key features of our implementation

- Allows easy configuration of the model, configurations are organized in JSON format thus makes the configuration legible to humans.
- Supports several types of data readers/writers.
- Enables us to dump CNN features for their use in other applications.
- Facilitates in loading pre-trained model and dumping the trained model.
- Supports two and three dimensional convolutional models.
- Run efficiently in CPU and GPU architectures.

Our implementation is publicly made available in github[1]. Sample configurations for some well known dataset like MNIST and CIFAR are also made available with it.

[1] `https://github.com/IITM-DONLAB/python-dnn`

CHAPTER 3

SAMPLE

3.1 Bibliography with BIBTEX

3.2 Other useful LATEX packages

Figure 3.1: Two IITM logos in a row. This is also an illustration of a very long figure caption that wraps around two two lines. Notice that the caption is single-spaced.

Table 3.1: A sample table with a table caption placed appropriately. This caption is also very long and is single-spaced. Also notice how the text is aligned.

x	x^2
1	1
2	4
3	9
4	16
5	25
6	36
7	49
8	64

APPENDIX A

A SAMPLE APPENDIX

Just put in text as you would into any chapter with sections and whatnot. Thats the end of it.

Publications

1. S. M. Narayanamurthy and B. Ravindran (2007). Efficiently Exploiting Symmetries in Real Time Dynamic Programming. *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 2556–2561.

REFERENCES

Amarel, S., On representations of problems of reasoning about actions. *In* **D. Michie** (ed.), *Machine Intelligence 3*, volume 3. Elsevier/North-Holland, Amsterdam, London, New York, 1968, 131–171.

Barto, A. G., **S. J. Bradtke**, and **S. P. Singh** (1995). Learning to act using real-time dynamic programming. *Artificial Intelligence*, **72**, 81–138.

Bellman, R. E., *Dynamic Programming*. Princeton University Press, 1957.

Crawford, J. (1992). A theoretical analysis of reasoning by symmetry in first-order logic. URL `citeseer.ist.psu.edu/crawford92theoretical.html`.

Griffiths, D. F. and **D. J. Higham**, *Learning LaTeX*. SIAM, 1997.

Knoblock, C. A., Learning abstraction hierarchies for problem solving. *In* **T. Dietterich** and **W. Swartout** (eds.), *Proceedings of the Eighth National Conference on Artificial Intelligence*. AAAI Press, Menlo Park, California, 1990. URL `citeseer.ist.psu.edu/knoblock90learning.html`.

Kopka, H. and **P. W. Daly**, *Guide to LaTeX (4th Edition)*. Addison-Wesley Professional, 2003.

Lamport, L., *LaTeX: A Document Preparation System (2nd Edition)*. Addison-Wesley Professional, 1994.

Manning, J. B. (1990). *Geometric symmetry in graphs*. Ph.D. thesis, Purdue University.

Ravindran, B. and **A. G. Barto** (2001). Symmetries and model minimization of markov decision processes. Technical report, University of Massachusetts, Amherst.