Video Event Localisation and Classification

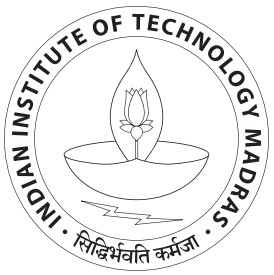
A Project Report

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]. Architecture of the indigenous DNN toolkit is shown on 2.1.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`

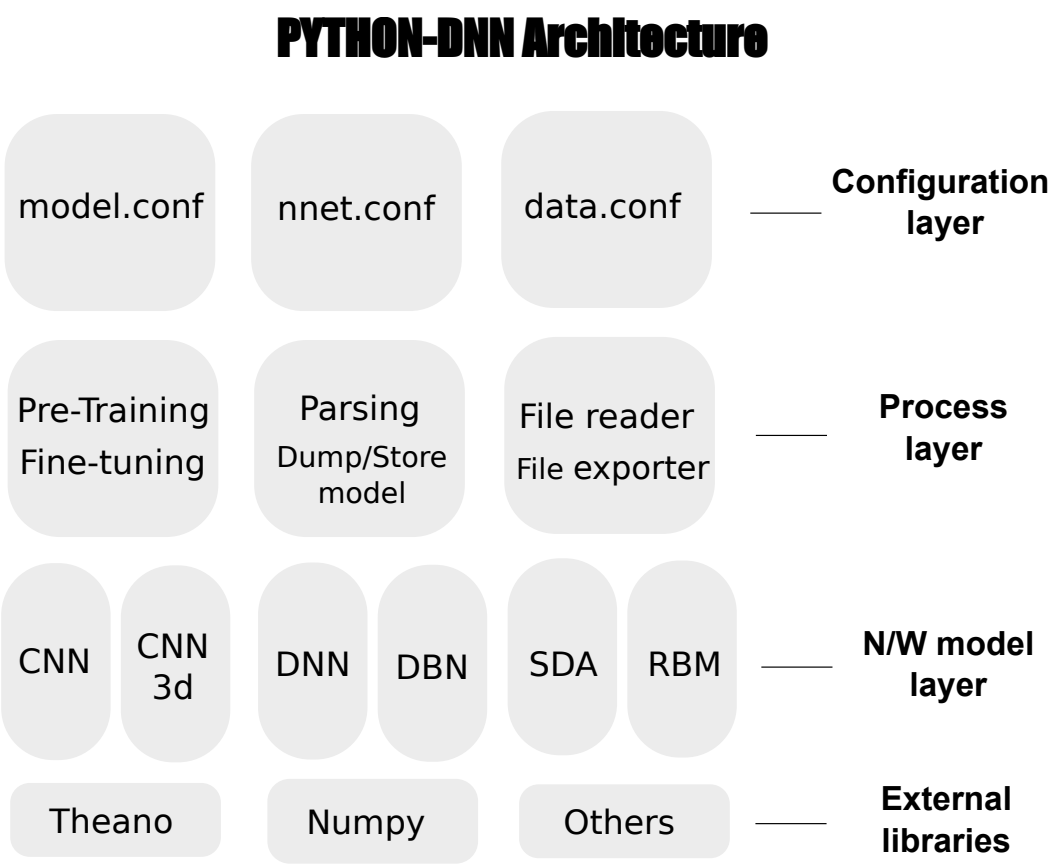


Figure 2.1: Architecture of Python-DNN

CHAPTER 3

BACKGROUND SUBTRACTION TECHNIQUES

In order to identify multiple events occurring simultaneously, we extract temporal volumes by bounding the visual field corresponding to the event. Extracted temporal volumes are fed to a three dimensional CNN to identify the corresponding event.

Event localization can be performed by determining the position with pixel changes and understanding the motion relativity as discussed in Basharat *et al.* [2008], where all components corresponding to an event show a similar flow of pixels. Following approach works significantly well in surveillance data but fails terribly in case human event detection. Later realized that, if we eliminate the background(static parts) of the visual frame, remaining would correspond to possible event locations. After background subtraction, we segment the visual frame and build temporal volumes corresponding to different events. Several background subtraction technique are eminent in the Piccardi [2004], among these some apply on static image while others on dynamic video. In case of our application we can blend both these approaches to generate more reliable event detectors.

3.1 Frame Differencing

A very common approach for performing the moving object segmentation is using frame differencing but these approach is very sensitive to small changes and yields

lots of noise when a camera is in motion. A simplest way to implement this technique is by computing the absolute difference of gray scale/intesity values two consecutive frames. Consider $P(p_x, p_y, t)$, intesity of pixel (p_x, p_y) at frame t, then pixel (p_x, p_y) that is considered foreground when absolute difference is above a threshold,

$$|P(p_x, p_y, t+1) - P(p_x, p_y, t)t| > T$$

.

The robustness of this method depends on speed of forground elements. Faster movements may require higher thresholds to reduce the noise. An alternate approach is to replace difference of a single previous frame by an average of multiple previous frames, still results are noisy and not reliable.

3.2 Eigen subtraction

An alternate approach named eigen background subtraction was seen to be more elegant. A sample of N images of the videos are obtained, mean background image v_b is computed and all mean normalized images X. Then we perform principal component analysis on the mean normalized images, idea is, that a high-dimensional images are often described by correlated variables and only a few meaningful dimensions account for most of the information. But performing PCA is not straightforward, consider 100 images of dimension 100x100 pixels, then the dimension of generated covariance matrix would be 10000x10000, i.e roughly 0.8 GB (considering 64 bit float values). Solving this is not feasible, hence we apply a trick from linear algebra that for a MXN matrix with $M > N$, we can at most have N-1 non-zero eigen values Duda *et al.* [2000]. So we perform eigen decomposition

of $X^T X$:

$$X^T X v_i = \lambda_i v_i$$

$$XX^T(Xv_i) = \lambda_i(Xv_i)$$

Hence the orthonormal eigen vector of XX^T is found out by normalizing Xv_i to unit length. Once eigen vectors of XX^T are computed, we project the current frame F on the eigen vector with top eigen values, and the reconstructed frame F' is obtained by using the projection coefficients and the eigen vectors. The difference $F - F'$ would correspond to the mask on the moving object.

3.3 Mixture of Gaussian

This is one of the well known method of extracting foreground information. Kaew-TraKulPong and Bowden [2002] model each background pixel by a mixture of N Gaussian distributions. The weights of the mixture represent the time proportions that those colors stay in the scene. The probable background colors are the ones which stay longer and more static. At any instance i a particular pixel (p_x, p_y) has color space represented as $V(p_x, p_y, i)$, history of the pixels are given by

$$X_1, .., X_t = V(p_x, p_y, i) \ : \ 1 \leq i \leq t$$

The history is modeled by mixture of N gaussian mixtures,

$$P(X_t) = \Sigma_i^N w_{(i,t)} \eta(X_t, \mu_{(i,t)}, \Sigma_{(i,t)})$$

$w_{(i,t)}$,$\mu_{(i,t)}$,$\Sigma_{(i,t)}$ are weight,mean and covariance of the i^{th} Gaussian in the mixture

at time t respectively and where η is gaussian density function. General principle behind this approach is that when a new object occludes the background object, it will not match one of the existing distributions. Instead willl result in either the creation of a new dis-tribution or the increase in the variance of an existing distribution. The variance of the moving object is expected to remain larger than a background pixel until the moving object stops.

Results of foreground mask obtained using moving average and eigen subtraction methods are shown in Figure **??**.

CHAPTER 4

Saliecny Detection

Visual saliency is the distinct subjective perceptual quality which makes some items in the world stand out from their neighbors and immediately grab our attention. As we had discussed earlier about the different techniques to extract the eminent pixels which are in motion, sometimes pixels which are stationary and quite distinct might also play a important role in the the activity recognition. These approaches are considered during the process attention modelling. Most of saliency estimations techniques in literature have got a common process,

1. **Color equalization:** It is done by removing the tail in the color histogram. This helps to improve the contrast in an image. In our implmentation color equalization is done on each channel separetely so that all channels can be provided to the clustering algorithm.
2. **SLIC-based segmentation:** It is a simple linear iterative clustering that clusters pixels in the combined five-dimensional color and image plane space to efficiently generate compact, nearly uniform superpixels. Any slic based clustering algorithm takes two parameters number of super-pixels and compactness of each superpixel. According to [Achanta *et al.*, 2012], slic is fastest and most memory efficient compared to other segmentation technique like graph-cut, quick shift segmentation technques. Segmentation reduces the number of computation in the subsequent stages
3. **Extract segment properties:** Extracting features of the region that are used in computing the saliecny of each regions. Some features that are considered in our implementation are color based features (LAB,RGB) and texture based features (GLCM,Texture Flow)
4. **Compute saliency:** Saliecny estimation techniques can be broadly categprozed into bottom-up, top-down and information maximization based on algorithmic computation. Earliest saliency based attention modelling was proposed by Itti *et al.* [1998]. It was inspired by the behavior and the neuronal architecture of the early primate visual system.

5. **Saliency cut:** Saliency maps are generally thresholded to obtain the salient mask. In the implementation we applied the grab-cut segmentation using the saliency estimation to obtain the salient mask. According to

Some of the distiguished saliency estimation techniques are discussed below.

4.1 Color Frequency Based

4.2 Contxt Aware Based

4.3 Spectral Distribution Based

4.4 Regional Contrast

CHAPTER 5

SAMPLE

5.1 Bibliography with BIBTEX

5.2 Other useful LATEX packages

Table 5.1: 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

REFERENCES

Achanta, R., **A. Shaji**, **K. Smith**, **A. Lucchi**, **P. Fua**, and **S. Susstrunk** (2012). Slic superpixels compared to state-of-the-art superpixel methods. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **34**(11), 2274–2282.

Basharat, A., **Y. Zhai**, and **M. Shah** (2008). Content based video matching using spatiotemporal volumes. *Comput. Vis. Image Underst.*, **110**(3), 360–377. ISSN 1077-3142. URL `http://dx.doi.org/10.1016/j.cviu.2007.09.016`.

Duda, R. O., **P. E. Hart**, and **D. G. Stork**, *Pattern Classification (2Nd Edition)*. Wiley-Interscience, 2000. ISBN 0471056693.

Itti, L., **C. Koch**, and **E. Niebur** (1998). A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on pattern analysis and machine intelligence*, **20**(11), 1254–1259.

KaewTraKulPong, P. and **R. Bowden**, An improved adaptive background mixture model for real-time tracking with shadow detection. *In Video-based surveillance systems*. Springer, 2002, 135–144.

Piccardi, M., Background subtraction techniques: a review. *In Systems, Man and Cybernetics, 2004 IEEE International Conference on*, volume 4. 2004. ISSN 1062-922X.