Temporal Action Recognition for Tutorial Videos

Jiaqi Fei

A report submitted for the course
COMP 8755: Individual Project
Supervised by: Zhenchang Xing
The Australian National University

October 2019

© Jiaqi Fei 2019
Except where otherwise indicated, this report is my own original work.

Jiaqi Fei
21 October 2019
Acknowledgments

This project provides me an opportunity to explore more on computer vision, and further practice what I learnt in ENGN 6528. Having experienced the whole process and reviewing a lot of academic papers, I have greatly developed my ability on researching, and had a better understanding on relative fields. Therefore, I sincerely appreciate my supervisor, Dr. Zhenchang Xing for allowing me participate in this project. He always encourages me, helps me contact others for help and guides me to the right direction. Meanwhile, I would like to thank the Phd student, Dehai Zhao, of my supervisor. He spares a lot of time to lead me finish the project, and help me analyse and solve some technical issues. At last, I wish to appreciate all other persons who ever provided me help, including Prof. Weifa Liang for answering me a lot of questions on course enrollment, and CECS IT help desk for providing VPN access.
Abstract

While design works relying on softwares such as Photoshop and InDesign become more popular, a great number of tutorial videos are uploaded for guiding new learners and inspiring designers. With the rapid development in artificial intelligence and breakthroughs in deep learning, currently it is possible for computers to learn from videos consisting of valid consequent designing activities and provide hints of possible next step for designers. This requires analysing temporal information besides recognising each activity in a video. Actually, many research groups, inspired by related works such as capturing motion track optical flows, 3D-CNN and object detection approaches such as Faster-RCNN, begin to work on temporal video recognition. In this thesis, I construct a brand new dataset consisting of manually-labelled frames extracted from tutorial videos of design-related software, and validate based on R-C3D model. Through experiment, I found that dataset consisting of videos with multiple labels will easily result in overfitting in training process, and training network obviously degraded with deeper training. Therefore, I replaced feature extraction model with ResNet and found it greatly improved the performance.
Contents

Acknowledgments iii
Abstract v

1 Introduction 1
  1.1 Problem Statement ............................................. 1
  1.2 Motivations ...................................................... 2
  1.3 Contribution ..................................................... 2
  1.4 Report Outline ................................................... 2

2 Background and Related Work 5
  2.1 Background ....................................................... 5
  2.2 Related work ..................................................... 6
    2.2.1 Object detection (region proposal method) ............... 6
    2.2.2 Convolutional 3D neural network (C3D) ................. 7
    2.2.3 Slow-fast network ......................................... 7
    2.2.4 Structured segment networks (SSN) ....................... 8
    2.2.5 R-C3D ...................................................... 9
  2.3 Summary ........................................................ 10

3 Design of Dataset 13
  3.1 The Main Manually Labelled Dataset of Tutorial Videos .... 13
    3.1.1 Overview of Dataset ...................................... 13
    3.1.2 Potential Issues ........................................... 13
  3.2 Backup Data Supplement Method ................................ 16
  3.3 Summary ........................................................ 17

4 Experimental Methodology 19
  4.1 Methodology ..................................................... 19
    4.1.1 3D-Net ...................................................... 19
    4.1.2 proposal Generation ...................................... 20
    4.1.3 Non-Maximum Suppression ................................ 21
    4.1.4 Classification ............................................. 21
    4.1.5 Optimisation .............................................. 21
  4.2 Experiment ..................................................... 22
    4.2.1 Experiment Setting ....................................... 22
    4.2.2 Data Preparation ......................................... 23
## Contents

4.2.3 Evaluation .................................................. 23  
4.3 Summary .................................................... 24  

5 Results  
5.1 Experiment with C3D .......................................... 25  
5.2 Experiment with 3D-Res34 ................................. 26  
5.3 Summary of Result ........................................... 26  

6 Conclusion  
6.1 Future Work ................................................ 29  

Bibliography ....................................................... 31  

Appendix: Study Contract ...................................... 33
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Tasks of temporal video recognition [Koprinska and Carrato, 2001]</td>
<td>2</td>
</tr>
<tr>
<td>2.1</td>
<td>Different outputs from 2D convolution and 3D convolution [Tran et al., 2015]</td>
<td>7</td>
</tr>
<tr>
<td>2.2</td>
<td>Overall architecture of C3D [Tran et al., 2015]</td>
<td>7</td>
</tr>
<tr>
<td>2.3</td>
<td>Overall architecture of Slow-fast network [Feichtenhofer et al., 2018]</td>
<td>8</td>
</tr>
<tr>
<td>2.4</td>
<td>Overall architecture of SSN [Zhao et al., 2017]</td>
<td>9</td>
</tr>
<tr>
<td>2.5</td>
<td>Overall architecture of R-C3D network [Xu et al., 2017]</td>
<td>10</td>
</tr>
<tr>
<td>3.1</td>
<td>A simple rotation activity consisting of three video frames</td>
<td>14</td>
</tr>
<tr>
<td>4.1</td>
<td>Architecture of Resnet block</td>
<td>20</td>
</tr>
<tr>
<td>4.2</td>
<td>Proposal Generation Subnet [Xu et al., 2017]</td>
<td>21</td>
</tr>
</tbody>
</table>
List of Tables

2.1 Comparison of mAP between different methods on Thumos 14 . . . . . . . 6
3.1 Class list of generated dataset . . . . . . . . . . . . . . . . . . . . . . . . . . . 15
5.1 C3D: test result with learning rate at 0.001, narrow proposal scale . . . 26
5.2 C3D: test result with learning rate at 0.0001, narrow proposal scale . . 26
5.3 C3D: test result with learning rate at 0.0001, larger proposal scale . . . 27
5.4 3D-Res34: test result with learning rate at 0.0001, larger proposal scale . 27
Chapter 1

Introduction

1.1 Problem Statement

Currently, richer Internet life include growing human activities on the Internet, such as coding, designing and browsing. If we want to learn from human activities to improve performance of human online behaviours, videos recording a sequence of human motions performed on computers will be an effective material. Taking designing work as an example, top design-related videos have millions of views in total. People demonstrate a desire to learn about design activities step by step. This brings a challenge for computer vision to automatically analyse those videos and classify video segments. More specifically, we want to know how different activities move following the timeline in a video. This demands recognising frame relationship of a video and classify activities.

An example of similar works could be recognising irregular activities in videos [Boiman and Irani, 2007]. These tasks all require a descriptor for videos, describing objects, activities or rules within a video. However, there are obvious limits on existed methods, such as assuming only a single object or activity existed in the video [Montes et al., 2016], or relying on pre-defined models to match motions [Heisele et al., 1997]. Meanwhile, while old methods attempt to segment and recognise on single frame, temporal coherence will be broken. Even though there is only minor change between two consequent frames, only relying on matching spatial features ignore the possibility that spatial boundary changes over time [Grundmann et al., 2010].

Therefore, recent tasks pursuing more accurate results give consideration to both of the temporal and spatial dimensions of a video. Figure 1.1 reflects a basic architecture of how to combine both temporal and spatial information. Overall, we wish to segment a video to obtain a series of consequent activities annotated by start-end time, so that we can learn about how designers conduct design activities, and in future, we can conclude to provide hints for improving human design behaviors.
Introduction

Figure 1.1: Tasks of temporal video recognition [Koprinska and Carrato, 2001]

1.2 Motivations

Nowadays, some video websites enable audience to know the start time of some events happened in a video, so that audience is able to conveniently skip to what they expect to watch. However, current approaches mostly rely on manual label during content review. Additionally, new designers who want advice on their stagnant work need to look through the whole tutorial video. This is time-consuming while a good result is not guaranteed. Therefore, we hope to have the computer automatically learn about professional activities in design work and analyse the sequence. By doing this, the computer may be able to predict and provide hints for the next step when people are stuck.

1.3 Contribution

The main contributions consist of three parts:
1. We create a new dataset, consisting of around 10000 labelled frames extracted from tutorial videos. Different from mainstream benchmark datasets such as Thumos14, ActivityNet and Charades, this dataset provides a series of new activities based on human design works.
2. As current researches mainly based on benchmark dataset, we experiment our own dataset on existed models to validate the efficiency and applicability. We provide a comparison between varying hyperparameter settings.
3. Through results obtained by further experiment design, we find the applicability and provide possible improvements.

1.4 Report Outline

Chapter 2 introduces several possible and related works. In this chapter, I briefly describe the architectures of each method, and provide a comparison to illustrate the performance.
Chapter 3 will introduce our own dataset, including an overview and examples. Some existed issues and possible solutions will also be discussed. Chapter 4 will present the methodology implemented in this project in details, and related experiment settings. Chapter 5 demonstrates an overview of the evaluation results obtained from designed experiments. Specific data will be provided to help understand each performance. Chapter 6 gives a conclusion of what I discovered from the research, and conclude improvements that can be made in the future.
Chapter 2

Background and Related Work

This chapter will demonstrate an overview of reviewed relevant researches, and contrast the performance of varying methods. Section 2.1 gives a skeleton and simple contrast of reviewed methods. Related work is briefly introduced in Section 2.2.

2.1 Background

Based on current researches, there are three main categories of approaches for temporal activity detection [Buch et al., 2017].

One is firstly exploiting a 3D convolutional neural network to extract feature maps, and then directly train the feature maps with recurrent neural networks (RNN), such as long short-term memory (LSTM) to explore temporal correlations. According to recent experiment, temporal activity localization performed on Thumos 14 with LSTM can achieve 58.74% mAP with a 0.5 IoU (Intersection over Union) threshold [Montes et al., 2016]. However, LSTM is not an end-to-end training and based more on sequences, thus current research exploiting LSTM only works for videos with single activities. This is the main reason that this method is not appropriate for our dataset. Additionally, LSTM requires vast resources for complex computation, which will bring huge load to training devices.

The second and the most recognized approach exploits proposal generation over feature maps extracted from C3D for classification. Some typical methods include R-C3D which consists of faster R-CNN and C3D, and structured segment networks which introduces optical flows. Most efficient methods of this category can achieve an approximately 30% mAP over a 0.5 IoU threshold.

And the last one is called SS-TAD [Buch et al., 2017], aiming to pursuit faster and simpler training. Based on feature extraction by C3D, it introduces two separate modules for temporal information detection and feature classification respectively. This network also generates proposals for extracting spatial features and finding regions of activities. However, it relies on other models such as SSD [Liu et al., 2016] and YOLO [Redmon et al., 2016], which exploits regression method for proposal generation instead of CNN.

IoU mentioned above is an evaluation method to measure how detected areas match
Table 2.1: Comparison of mAP between different methods on Thumos 14

<table>
<thead>
<tr>
<th>Model</th>
<th>mAP at 0.5 IoU, on Thumos 14</th>
<th>mAP, on Charades</th>
</tr>
</thead>
<tbody>
<tr>
<td>RNN [Montes et al., 2016]</td>
<td>&gt; 50%</td>
<td></td>
</tr>
<tr>
<td>R-C3D [Xu et al., 2017]</td>
<td>28.9%</td>
<td>12.7%</td>
</tr>
<tr>
<td>SSN [Zhao et al., 2017]</td>
<td>29.8%</td>
<td></td>
</tr>
<tr>
<td>Faster R-C3D [Wang and Cheng, 2018]</td>
<td>38.5%</td>
<td></td>
</tr>
<tr>
<td>SS-TAD [Buch et al., 2017]</td>
<td>29.2%</td>
<td></td>
</tr>
</tbody>
</table>

the ground truth. This can be mathematically represented as:

\[
\text{IoU} = \frac{\text{DetectionResult} \cup \text{GroundTruth}}{\text{DetectionResult} \cap \text{GroundTruth}}
\]

As a general measurement, IoU is widely exploited by different models to evaluate performance, and can be a baseline for mutual comparison. Table 2.1 provides a comparison of mAP values based on the same benchmark dataset. In the table, mAP of RNN is only for videos with one single activity, and only R-C3D experimented on Charades dataset. It is obvious that temporal action detection with proposal method present advantages comparing to other approaches. Based on main relevant work from 2016 to 2018 that I reviewed, many approaches for temporal activity detection are based on generating proposals, and all perform relatively good mAP.

2.2 Related work

2.2.1 Object detection (region proposal method)

No matter that it is for an image or a video, the actual object to detect usually only occupy a small part of the whole picture. In object detection, different from classification, the main purpose focuses more on finding the difference between aimed object and all other elements [Papageorgiou et al., 1998].

Earliest work only works on rigid objects, as the method simply attempts to match objects with pre-defined templates [Betke and Makris, 1995]. Recent object detection models are more extensive as they capture the structural representation of an object, which is driven by region proposal method and RoI (region of interest) pooling.

To generate proposals, there are two main methods mentioned in section 2.1, one is convolutional networks motivated by selective search [Uijlings et al., 2013], which will be further discussed in this section, and the other is regression as mentioned above. Convolutional networks based on region proposal method are called region-based CNN (R-CNN). In region proposal method, the network will exploit sliding windows of varying sizes to mapping features from various areas. The following RoI pooling will pooling these features into small feature map for attempting to match
§2.2 Related work

Figure 2.1: Different outputs from 2D convolution and 3D convolution [Tran et al., 2015]

![Figure 2.1: Different outputs from 2D convolution and 3D convolution](image)

Figure 2.2: Overall architecture of C3D [Tran et al., 2015]

![Figure 2.2: Overall architecture of C3D](image)

Almost all reviewed methods mentioned above exploit C3D to extract temporal features from video frames. Activity detection requires recognizing a complete action from its starting to the ending frame. C3D can extracting both spatial and temporal features through a three-dimensional fully convolutional network. In traditional 2D convolution, the feature maps will be reduced to two dimensions. Shown as Figure 2.1, even for inputs with multiple channels, all the feature maps will be summed up to one final feature map. Therefore, a video through 2D convolution will output only one image, and the temporal information is lost [Tran et al., 2015]. However, different from 2D convolution, 3D convolution exploits convolution kernels that have one more dimension. In video recognition, inputs of an activity consisting of multiple frames can be regarded as a stacked cube, whose three dimensions represent height, width and timeline respectively. By performing convolution with a 3D kernel, the kernel will be able to read and preserve the temporal features from the temporal dimension. Another feature of C3D is that this architecture will firstly focus on the whole picture and then track objects with the most obvious changes [Tran et al., 2015]. Due to the motion over time, C3D can selectively record temporal changes of the main object.

The general architecture is presented as Figure 2.2. Similar to 2D convolution, it performs 8 convolutions and 5 poolings, followed by 2 fully connected layers. However, all the convolutional kernels and pooling kernels are 3-dimensional.

### 2.2.2 Convolutional 3D neural network (C3D)

Slow-fast network is also a work with good accuracy, considering temporal information of a video. As shown in Figure 2.3, there are two pathways consisting of different convolutional models and frame rates. The slow pathway can accept any kine of 3D convolutional models, but 3D kernels are only performed over limited...
Background and Related Work

Figure 2.3: Overall architecture of Slow-fast network [Feichtenhofer et al., 2018]

layers. Meanwhile, adopted frame rate for input video is relatively low. In contrast, the fast pathway adopts a high frame rate for the same raw clip of the input video. While features in different pathways have different sizes, two pathways are connected so that slow pathway can learn about how features are extracted in the fast pathway, and match the feature size before final classification. However, experiment of Slow-fast network is mainly set for activity classification. Therefore the final outputs will provide predictions on activities included in a sequential of frames, instead of segmenting videos into a series of activities.

2.2.4 Structured segment networks (SSN)

To detect actions in untrimmed videos, the paper [Zhao et al., 2017] exploits a structured segment network with proposal method to both classify actions and verify the completeness. There are basically three main steps in this framework. Firstly, the framework will generate augmented proposals of different time fragments. In this step, both the starting and ending of candidate regions provided by raw data will be extended by half of the original duration. Shown as Figure 2.4, the region in green box will be expanded to the yellow box.

The next step is performing structured temporal pyramid pooling over the generated proposals. Based on generated proposals, this network exploits sparse snippet sampling to generate temporal pyramid for the three stages. This method will evenly divide several segments from the each proposal, and randomly select 1 snippet from each segment. Here, each snippet consists of original images and their optical flows, which reflects the speed and direction of pixel motion. Meanwhile, the network will split all the proposals into three stages. The three stages include starting, course and ending, which respectively record the start, the process and the end of each action. Therefore, to build a stage-level temporal pyramid of arbitrary level, the framework

https://blog.postmarkapp.com/0000
extracts feature vectors of every single snippet in each stage, and concatenates pooled feature vector of each level into one pyramidal representation. The level here will still evenly divide a stage into different parts. Similarly, the overall proposal-level feature representation will be built by concatenating all the three stage-level temporal pyramids into one complete representation. By performing proposal augmentation and STPP, the later training step can learn from both the proposal and its context [Zhao et al., 2017].

Lastly, an activity classifier and a completeness classifier will respectively recognize actions and evaluate the completeness. Similar to other networks with proposal pooling, the classifier here will classify inputs into K action classes plus 1 background class. Proposals are predicted to be representations will be kept and combined, while those considered as background will be removed.

### 2.2.5 R-C3D

R-C3D is a network directly exploiting Faster R-CNN architecture for temporal recognition. Figure 2.5 presents the overall architecture of R-C3D. This network consists of a feature extractor using C3D, a proposal generation method and the final classification stage.

As mentioned in section 2.2.2, 3D convolutional network is significant for extracting both spatial and temporal features in video buffer. Therefore, R-C3D still firstly relies on 3D convolutional network. Here, the outputs of the feature extraction include information of video frames, which represent temporal information, and video height and width, which represent the spatial information. The generated feature maps will be shared by other subnets to ensure the efficiency of computation. Meanwhile, R-C3D avoids recurrent layers by encoding video information with a 3D fully-convolutional network.

Generated feature maps will enter proposal subnet. The proposal generation stage
introduces sliding windows of varying sizes to find different candidate proposals. However, proposals generated by proposal subnet may demonstrate low confidence and overlap with each other. Under this circumstance, this model exploits NMS (non-maximum suppression) to select valid proposals with appropriate confidence. For the fully-convolutional layer before final classification, selected proposals will go through a 3D RoI pooling to make all the feature maps have the same fixed size. The final classification task consists of two layers, which classify the specific category and starting and ending time respectively. As R-C3D is the baseline method of this project, more specific implementation details of R-C3D will be discussed in Chapter 4.

2.3 Summary

In this chapter, we introduce some basic architectures related to extracting 3D features, and describe how several possible networks works for video activity recognition. Some crucial basic architecture include C3D and proposal generation method, which achieve spatio-temporal information extraction and object detection respectively. Meanwhile, by comparing different networks, we choose R-C3D for this project, considering accuracy, speed and the aim of temporal recognition. Overall, while Slow-
fast network focus more on classification, R-C3D presents the highest accuracy among else networks, as shown in Table 2.1. Additionally, in our dataset, each video contains multiple varying activities, which means RNN method is inappropriate. And only R-C3D experiments on a similar dataset, Charades. Therefore, we finally choose R-C3D as a baseline method, and revise it to fit our dataset to validate its feasibility. This will be discussed in Chapter 4.
Background and Related Work
Chapter 3

Deisgn of Dataset

In this chapter, data-related work will be demonstrated. Section 3.1 will present the dataset finally adopted in training stage, including the classes list generated from the dataset, and Section 3.2 will introduce a backup method for solving potential issues existing in the dataset.

3.1 The Main Manually Labelled Dataset of Tutorial Videos

3.1.1 Overview of Dataset

Our data resource is tutorial videos on YouTube, uploaded by varying authors. The general fps of original videos is 30 frames per second. By selecting one frame from every second, I finally obtained around 10,000 frames of different activities. All the frames will be numbered in accordance with the temporal order they are extracted. Therefore, extracted consecutive frames can basically represent the original sequential activities. Figure 3.1 presents three consecutive frames of one rotation activity. The position changes demonstrated in Figure 3.1(a), 3.1(b) and 3.1(c) reflect the temporal changes.

From the 10,000 frames, we extracted 21 most common activities, and 1 ambiguous category containing all the irrelevant and indefinable operations, including the starting and the ending of a video, long-time pause and vague activities. The whole class list of activities is listed as Table 3.1.

3.1.2 Potential Issues

As a manually labelled dataset, there will be foreseeable issues such as wrong labels for some activities and vague definition on classification. Additionally, mistakes made by designers will lead to inconsecutive motions presented in videos. All the potential issues will be discussed in this section.

First of all, as mentioned above, manually labelling may lead to inconsistency and mistakes. The main reason is that it hard to assure an accurate understanding on the activities performed by the authors. Additionally, there may be several methods leading to the same designing goal. This may make same activities be labelled differently.
Design of Dataset

Figure 3.1: A simple rotation activity consisting of three video frames
### §3.1 The Main Manually Labelled Dataset of Tutorial Videos

#### Table 3.1: Class list of generated dataset

<table>
<thead>
<tr>
<th>Class Number</th>
<th>Class Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Element Addition</td>
</tr>
<tr>
<td>2</td>
<td>Element Remove</td>
</tr>
<tr>
<td>3</td>
<td>Element Cut</td>
</tr>
<tr>
<td>4</td>
<td>Element Motion</td>
</tr>
<tr>
<td>5</td>
<td>Element Rotation</td>
</tr>
<tr>
<td>6</td>
<td>Element Stretch</td>
</tr>
<tr>
<td>7</td>
<td>Text Addition</td>
</tr>
<tr>
<td>8</td>
<td>Text Modification</td>
</tr>
<tr>
<td>9</td>
<td>Text Font Size Setting</td>
</tr>
<tr>
<td>10</td>
<td>Text Remove</td>
</tr>
<tr>
<td>11</td>
<td>Text Motion</td>
</tr>
<tr>
<td>12</td>
<td>Drop-down Box Setting</td>
</tr>
<tr>
<td>13</td>
<td>Zoom Up</td>
</tr>
<tr>
<td>14</td>
<td>Zoom Down</td>
</tr>
<tr>
<td>15</td>
<td>Colour Setting</td>
</tr>
<tr>
<td>16</td>
<td>Property Setting</td>
</tr>
<tr>
<td>17</td>
<td>Layer Switch</td>
</tr>
<tr>
<td>18</td>
<td>Layer Setting</td>
</tr>
<tr>
<td>19</td>
<td>Board Motion</td>
</tr>
<tr>
<td>20</td>
<td>Open External File</td>
</tr>
<tr>
<td>21</td>
<td>Ambiguous</td>
</tr>
</tbody>
</table>
Meanwhile, human cannot label the starting and ending time with a high precision, while then network will have a different representation of predicted time. The second issue is also caused by some vague presentation of the activities. To achieve the same effect, there may be different valid activities. For example, to change the font size, designers can either stretch the text box or directly rely on drop-down box to select another font size. Another aspect is that to finish an valid activity, there may be other valid activities existing in the duration. However, it will be meaningless to segment the whole duration into two activities. Therefore, this may also negatively affect the training.

Similarly, many activities may present the same effect, which may also bring vagueness to the training network. For example, when an element is set to the same color as the background, it will look like being removed. And by simply learning from frames, it will be hard to recognise the actual activity.

Additionally, in those videos, there are too many trivial but common operations. For example, in many activities, designers will need to scale up and scale down the whole interface for checking the effect. Currently, we are trying to subdivide each class to avoid too many common activities existing among different activities. However, this approach will result in several activity classes that are actually irrelevant to interface design. In following work, it can be considered to remove those categories for better training.

Lastly, some significant classes may only contain a relatively small quantity of frames. For example, while element remove is a significant activity in interface design, quantity of relevant frames extracted from videos is relatively small. We have done some work on this issue. And in the next section, a relevant solution to this issue will be introduced.

### 3.2 Backup Data Supplement Method

For the final potential issue mentioned in section 3.1, we attempt to automatically record the human behaviours on computer, aiming to design a method to help generate auto-labelled dataset. We rely on Python script to send execution commands to the operating system and simultaneously record the whole process. Resulted dataset can help supplement data of significant classes, which only have a few frames. An example of element remove is provided in the final paragraph of section 3.1.2. Under this circumstance, work of this stage can be exploited as a supplement. However, while this method will provide more data for the network to better learn about each activity, it may destroy the temporal information of original dataset. Thus we have this method as a backup, and this paper will mainly validate the dataset illustrated in section 3.1.
3.3 Summary

This chapter firstly demonstrates an overview of our own dataset, including data resources, frame segmentation, generated classes and potential issues. We obtained original videos by different authors from Youtube, and extract frames with regular intervals to represent a relatively static but comprehensible motion of the whole video. Then, we provide the result of manually labelling and discuss about the possible issues and related considerations. In section 3.2, focused on the data quantity issue mentioned previously, a relevant solution is introduced. More considerations and possible improvements will be discussed in section 6.1. And for the next chapter, specific experiment methodology will be presented.
Design of Dataset
Experimental Methodology

This chapter will firstly briefly introduce the R-C3D network we finally implement to train and validate our dataset in section 4.1. Then, in section 4.2, specific experiment setup and processes will be presented, including experimental environment in section 4.2.1, data preprocessing in section 4.2.2, and evaluation method in section 4.2.3.

4.1 Methodology

4.1.1 3D-Net

In experiment stage, I found the dataset is sensitive to overfitting. Therefore, eventually I validate our dataset on two different convolutional networks. One is general C3D and the other is 3D-Res34, which develops from ResNet34. While general architecture of C3D is introduced in section 2.2.2, this section will focus on illustrating 3D-Res34.

Each learning block is shown as Figure 4.1. For ResNet34, each block has two convolutions performed by a 3D kernel with the size of $3 \times 3 \times 3$, followed by a batch normalisation and a Relu. Input will go through an identity mapping, and be added to output of each block if they have the same dimensions. Otherwise, there will be two general methods to make two dimensions match. The first method is to increase the dimension by zero-padding, and the other is performing a $1 \times 1$ convolution to adjust the dimensions. However, the second method will bring new hyperparameters for network and increase the computation. Meanwhile, for the first method, while zero-padding may destroy the original distribution of features, a pooling with kernel size of $1 \times 1$ and stride of 2 can be exploited as downsample to maintain the original construction.

Input for each 3D-network will be a sequence of extracted frames with the size of $3 \times L \times 112 \times 112$, where 3 represent three channels of RGB, L represents the total number of input frames and two 112 are height and width of each frame. The output feature map will be at the size of $512 \times \frac{8}{7} \times 7 \times 7$. 
4.1.2 proposal Generation

The proposal generation method is based on the baseline architecture of R-C3D, shown as Figure 4.2. Assuming for each location on the temporal dimension of the sharing feature map extracted from last network, we will firstly have K anchor segments (like a candidate box containing a segment of the feature map) and their corresponding length. For each anchor segment, we need a score indicating how much one segment can be believed as an activity or background, and a relative offset on the temporal dimension to represent its location. As this is only related to temporal dimension, a temporal only feature map will be more convenient [Xu et al., 2017]. This requires the original sharing feature map firstly go through a $3 \times 3 \times 3$ convolution to obtain a more robust feature map. Next, a $1 \times 7 \times 7$ max-pooling will be implemented over the new feature map to obtain a $512 \times \frac{1}{8} \times 1 \times 1$ feature map, which only contains temporal information. Finally, two $1 \times 1 \times 1$ convolution will be implemented over the temporal only feature map to obtain two representations of all anchor segments. One is the score indicating the confidence of a segment being an activity or a background. And for the other, the 512-dimensional feature of each point on the $\frac{1}{8}$ will be used to predict relative offset and length. Positive samples are those segments overlap with ground truth with IoU larger than 0.7, while negative samples are those with IoU less than 0.3. At the last of this stage, the network samples from all the segments with 1:1 positive-negative ratio as final training batch. And the sampled anchor segments with its binary representation will be final proposals.
4.1.3 Non-Maximum Suppression

For proposals generated in the previous stage, there will be highly-overlapped and low-confidence proposals. Therefore, NMS is exploited here to remove those meaningless proposals. As higher confidence means a proposal is more likely to accord with prediction, proposals whose confidence are local maximum values will be selected. Among those selected proposals, find all groups of highly-overlapped proposals (IoU larger than 0.7), and for each group, just keep the one with maximum confidence.

4.1.4 Classification

A 3D RoI pooling will be implemented before fully-connected layers, which obtain feature map that can be used for classification. The pooling is for transferring arbitrary sizes of proposals into a fixed size, or the fully-connected layers will need convolutional kernels of different sizes. The pooling kernel is $\frac{1}{8} \times 4 \times 4$, which leads to $512 \times 1 \times 4 \times 4$ feature map for final classification. The final classification also consists of two different fully-connected layers, which used for activity classification and start-end time regression. Input of these two layers is from output of previous fully-connected layer.

4.1.5 Optimisation

As there are two fully-connected layers exploited for different classification tasks, the loss function also consists of two different parts. For classification, a softmax loss function is exploited, while a L1 loss function is used for temporal part. The
Experimental Methodology

Combination is shown as Equation 4.1 [Xu et al., 2017]:

$$\text{loss} = \frac{1}{N_{cls}} \sum L_{\text{softmax}}(a_i, a_i^*) + \frac{1}{N_{time}} \sum a_i^* L1(t_i, t_i^*)$$ (4.1)

where $N_{cls}$ is the batch size of training set, $N_{time}$ is the number of proposals. $a_i$ is the predicted probabilities of a proposal being an activity while $a_i^*$ is the ground truth. $t_i$ is predicted relative coordinate generated in proposal generation stage, while $t_i^*$ represents transformed coordinate from ground truth to relative offset represented in proposals. As we have obtained relative offsets in proposal generation stage, which include centre locations and lengths of segments, the transformation from ground truth to proposals can be calculated by Equation 4.2 [Xu et al., 2017]:

$$\begin{cases}
\text{center} = \frac{c_{gt} - c_p}{l_p}, \\
\text{length} = \log\left(\frac{l_{gt}}{l_p}\right)
\end{cases}$$ (4.2)

Here, center means how ground truth center is transformed to proposal center, length means how ground truth duration is transformed to proposal duration, $c_{gt}$ represents ground truth center while $c_p$ represents proposal center. And $l_{gt}$ and $l_p$ are ground truth duration and proposal duration respectively.

4.2 Experiment

4.2.1 Experiment Setting

Main working environment is based on Linux, with the implementation of Pytorch 0.4.0 and CUDA 9.0. The training process is deployed on a single Geforce Titan XP GPU. For all the frames describing valid and clearly-defined activities, we input 200 frames for each batch. A pre-trained model of ActivityNet based on Caffe is exploited.

This experiment consists of four training groups with varying settings, including different learning rates and proposal scales during training and different feature extraction networks. There are two groups of proposal scale settings, including a narrow one with [2, 4, 5, 6, 8, 9, 10, 12, 14, 16], and a large one with [2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56].

The first group of training exploits a starting learning rate of 0.001 and the narrow scale. The second group of training exploits a starting learning rate of 0.0001 and a narrow scale. The third group of training also starts with learning rate at 0.0001, but exploits a larger scale. And the fourth group of training also adopts a larger scale and 0.0001 as starting learning rate, but exploits 3D-Res34 as a feature extraction network. However, there are some fixed hyper-parameters. For example, learning decay step is 4 for all experiment groups, which means after the forth step of training iteration, the learning rate will reduce by 10 times.
4.2.2 Data Preparation

Each single video will be regarded as an input. Frames extracted from a video will be scaled down into a fixed size for the convenience of feature extraction. In each experiment, I have 70% of videos and their frames as training set, and the rest as testing set. All the classes except for "Ambiguous" are labelled from 1 to 20, and a single text file is generated for each class to only include all the frames presenting the same activity. For all the training videos, a text file called "segments.txt" will be generated to record all the activities going to be trained for each video.

Before training, both training and testing sets will be transformed to a format appropriate for reading in proposal information. Overall, this is a list whose elements are dictionaries. The key of each dictionary is the index of extracted segment. Its values include relative offset, duration, ground truth class, class corresponding to the maximum scores, maximum value of overlapping, whether frames are flipped, corresponding frames and their paths.

4.2.3 Evaluation

For evaluation, same as other object detection approaches, I adopt IoU and mAP here, as IoU is one demanded value for judging whether a prediction is correct. In R-C3D, the main task is to recognise whether there is an activity in generated proposals. Proposals in the model are temporal proposals, indicating a fragment from a starting time to an ending time extracted from the original video. Ground truth is labelled starting and ending time of a fragment indicating an activity.

Therefore, following Equation 2.1 of calculating IoU, the detection result here is the generated proposals with a label, ground truth is manual label. If IoU is calculated to be larger than a threshold, this proposal will be regarded as True Positive (TP), else is False Positive (FP). And precision is calculated as:

\[
\text{precision} = \frac{TP}{TP + FP}
\]  

Then, we can calculate average precision (AP) following Thumos14 challenge [Jiang et al., 2014] as:

\[
AP = \frac{\sum_{k=1}^{n} (P(k) \times \text{rel}(k))}{\sum_{k=1}^{n} \text{rel}(k)}
\]  

Here, \( n \) is the number of videos, \( P(k) \) is the precision at \( k \), \( \text{rel}(k) \) is 1 when the proposal is TP, otherwise it will equal to 0. Finally, mAP is calculated as:

\[
mAP = \frac{1}{C} \sum_{c=1}^{C} AP(c)
\]  

where \( C \) is the total number of all classes. In evaluation stage, AP value will represent the performance of each single class, and mAP will be regarded as a value indicating the overall performance.
4.3 Summary

In this chapter, specific experiment methodologies and settings are introduced. Section 4.1 gives a detailed implementation of the network. The whole network is based on R-C3D architecture. For feature extraction, I experimented the dataset on C3D and 3D-Res34 respectively. In details, this chapter illustrates the architectures of two 3D-networks, and how proposal generation subnet and NMS selects anchor segments and proposals. The optimisation combines a softmax loss function and a L1 loss function for two classification tasks.

Section 4.2 illustrates the design of experiments. There are four groups of experiments in total, comparing performances under varying learning rates and proposal scales. Before training, dataset will be segmented into different batches and transformed into a specific format. For evaluating performance, a mAP metric is introduced.
Results

This chapter will present the final result of our experiments. As our dataset is tested with varying experiment settings, there will be different sections presenting corresponding results. Results under C3d and 3D-Res34 will be respectively presented in section 5.1 and section 5.2, in each of which will also give results under varying learning rates and proposal scales during training process. At the end of this chapter, section 5.3 will give a brief summary of results and emerged issues.

Among the three benchmark datasets R-C3D experiments on, Charades is more similar to our dataset as it contains more complex activities and have multi-label data. Therefore, I exploit performance of R-C3D on Charades as a benchmark for comparison. In training and testing phases, I will remove some vague categories such as "Board motion" and "Ambiguous", and meaningless activities such as "Zoom up" and "Zoom down". By doing this, 10 categories with finer and meaningful activities are actually trained and tested. Though due to insufficiency of data quantity, some categories cannot be recognised, or just present a relatively low mAP, several common categories with enough data quantity support present a good performance, which is quite close to the benchmark. Overall, R-C3D can achieve the mAP at 12.7%, while the best test can reach mAP at 13.98%. Hence, the network can basically work on other complex dataset, and provide reference value for similar work.

5.1 Experiment with C3D

As mentioned in section 4.2.1, with C3D feature extraction method, I set up two groups with learning rate of 0.001 and 0.0001 and a narrow proposal scale, and one testing group with learning rate of 0.0001 and a larger proposal scale. In Table 5.1, I present the evaluation with regard to C3D in terms of learning rate at 0.001. There are only two categories are successfully recognised, with mAP at 8.25%. In comparison, evaluation with regard to C3D in terms of learning rate at 0.0001 shown in Table 5.2 has a 4% improvement, reaching mAP at 12.25%. As I want to check whether it appropriate to have the network learn from our data features rapidly, this reveals that for such dataset, overfitting should be firstly avoided. While I attempt to enlarge the proposal scales, mAP obtains a 2.05% improvement as presented in Table 5.3. Though there is slight improvement, the AP of each class fails to perform any significant
### Results

**Table 5.1:** C3D: test result with learning rate at 0.001, narrow proposal scale

<table>
<thead>
<tr>
<th>Category</th>
<th>AP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Colour Setting</td>
<td>7.4%</td>
</tr>
<tr>
<td>Property Setting</td>
<td>9.1%</td>
</tr>
<tr>
<td><strong>mAP</strong></td>
<td><strong>8.25%</strong></td>
</tr>
</tbody>
</table>

**Table 5.2:** C3D: test result with learning rate at 0.0001, narrow proposal scale

<table>
<thead>
<tr>
<th>Category</th>
<th>AP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Element Addition</td>
<td>6.9%</td>
</tr>
<tr>
<td>Colour Setting</td>
<td>17.6%</td>
</tr>
<tr>
<td><strong>mAP</strong></td>
<td><strong>12.25%</strong></td>
</tr>
</tbody>
</table>

Changes on performance.

#### 5.2 Experiment with 3D-Res34

While larger proposal scales can bring more choices for proposals with high confidence, and 0.0001 is a obviously better learning rate setting, I simply adopt these two settings as default setting in this stage. And due to insufficiency of some categories, the feature extraction method of this experiment phase will be changed to 3D-Res34 network with a deeper architecture. Table 5.4 presents the evaluated mAP with regard to 3D-Res34 in terms of learning rate at 0.0001. It is obvious the Average Precision (AP) for each recognised class also outperforms the result under C3D. Though the mAP presents slight decrease of 0.3%, it can be regarded as reasonable error. Due to that more activities can be recognised and performance on each recognised class can reach a acceptable AP, 3D-Res34 can be regarded as a better method. And from this phase of experiment, it can be seen that insufficient training of categories with small quantity is the main reason that some activities cannot be effectively recognised. However, with data support, the method can reach a acceptable result on such complex dataset, as the mAP is close and even higher than benchmark.

#### 5.3 Summary of Result

For categories with enough data quantity, the results can basically reach the level of benchmark, and are even 1-1.5% higher. From a series of experiments, we can find while the dataset contains complex features, it is quite sensitive to overfitting. Hence, to have the network fully learn from both the spatial and temporal features, a deeper residual network is a better approach. Meanwhile, enlarging the scale of anchors is
Table 5.3: C3D: test result with learning rate at 0.0001, larger proposal scale

<table>
<thead>
<tr>
<th>Category</th>
<th>AP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Colour Setting</td>
<td>17.4%</td>
</tr>
<tr>
<td>Property Setting</td>
<td>11.2%</td>
</tr>
<tr>
<td>mAP</td>
<td>14.3%</td>
</tr>
</tbody>
</table>

Table 5.4: 3D-Res34: test result with learning rate at 0.0001, larger proposal scale

<table>
<thead>
<tr>
<th>Category</th>
<th>AP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Element Addition</td>
<td>17.9%</td>
</tr>
<tr>
<td>Colour Setting</td>
<td>13.9%</td>
</tr>
<tr>
<td>Property Setting</td>
<td>9.8%</td>
</tr>
<tr>
<td>Open External File</td>
<td>14.3%</td>
</tr>
<tr>
<td>mAP</td>
<td>13.98%</td>
</tr>
</tbody>
</table>

beneficial for the training phase. However, issues emerge in the experiment phase. As mentioned in section 3.1.2, some significant activities have relatively small quantity of data, because these activities emerge rarely in tutorial videos. For example, element rotation is a highly recognisable activity from human vision. But in one video, this activity may only occur once, which lead to insufficiency of the data quantity. As a result, such activities cannot be effectively recognised because of insufficient training. However, for categories with sufficient data quantity, the performance is acceptable as the mAP is higher than benchmark, which shows the network can work on such dataset.
Results
Conclusion

This thesis introduces and validates an end-to-end method, R-C3D, for recognising temporal activities in tutorial videos. For extracting both the spatial and temporal information, C3D architecture is exploited. As for object detection, proposal generation method is also significant in this project. We manually labelled a dataset consisting of around 10000 frames extracted from tutorial videos. Two groups of experiments are set at initial stage to validate the performance under varying training environments. Focusing on the issue that our dataset is quite sensitive to overfitting, I designed another group of experiment exploiting ResNet to compare the evaluation results. From the experiment results, we can see that with pretrained models used, relatively low learning rate and deeper feature extraction architecture solving degradation can significantly improve the performance. And comparison in section 5.1 shows enlarging proposal scales have some reference significance for such dataset, while the influence scale of designing activities can be vague to identify. As the mAP can reach around 14%, which is higher than 12.7% of the benchmark, we can find the network is applicable to our dataset, and can provide reference value for similar works.

6.1 Future Work

As it is the first time I try to implement a complex research on temporal activity recognition independently, and due to limit on time and manpower, there are many shortages and issues discovered that cannot be tested and solved. Firstly, segmenting one long video into several short fragments is worth trying. This can help increase the frame rate, reduce irrelevant classes and have the network focus more on valid activities. Meanwhile, more work on data augmentation can be attempted. For example, each long activity can be segmented into several relatively short activities with same labels. Additionally, some meaningless activities and frames can also be extracted and labelled as background. This may help the network recognise static and dynamic frames, learn more about obvious changes which indicate meaningful activities and perform better when trying to find background proposals in training process. Finally, some other features of the video frames can be considered being incorporated into current network architecture. For example, optical flows can be generated to be trained with original frames.
Bibliography


Appendix: Study Contract
INDEPENDENT STUDY CONTRACT
PROJECTS

Note: Enrolment is subject to approval by the course convener

SECTION A (Students and Supervisors)

<table>
<thead>
<tr>
<th>UniID:</th>
<th>u6332860</th>
</tr>
</thead>
<tbody>
<tr>
<td>Surname:</td>
<td>Fel</td>
</tr>
<tr>
<td>First Names:</td>
<td>Jingl</td>
</tr>
<tr>
<td>Project Supervisor (may be external):</td>
<td>Dr. Zhenchong Xing</td>
</tr>
<tr>
<td>Formal Supervisor (if different, must be an HEWCP academic):</td>
<td></td>
</tr>
<tr>
<td>Course Code, Title and Units:</td>
<td>COMP 755, Individual Computing Project, 12 units</td>
</tr>
</tbody>
</table>

Semester: [ ] S1 [ ] S2 Year: [ ]

Two-semester project (12u course only): [ ]

Project Title:
Vision-based video analysis

Learning Objectives:
1. Have advanced understanding on computer vision and data processing
2. Can apply appropriate computing skills to a specific area and solve real-world challenges
3. Have a clear demonstration on progress of the project and relevant skills

Project Description:
1. Research about literature in computer vision and deep learning topic;
2. Learn to use TensorFlow or PyTorch, and relevant algorithms such as CNN for image and action recognition;
3. Collect data from observing developers’ actions, pre-process data to generate developers’ workflow;
4. Analyse how developer works and design assessment, aiming to build the connection between software-related entities and developers’ behaviours, and develop a knowledge base accordingly;
5. Write report.

Research School of Computer Science

Form updated Nov 2017
ASSESSMENT (as per the project course’s rules web page, with any differences noted below).

<table>
<thead>
<tr>
<th>Assessed project components:</th>
<th>% of mark</th>
<th>Due date</th>
<th>Evaluated by:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Report: name style:</td>
<td>(min 45%)</td>
<td></td>
<td>(examiner)</td>
</tr>
<tr>
<td>(e.g. research report, software description...)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Artifact: name kind:</td>
<td>(max 45%)</td>
<td></td>
<td>(supervisor)</td>
</tr>
<tr>
<td>(e.g. software, user interface, robot...)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Presentation:</td>
<td>(10%)</td>
<td></td>
<td>(course convener)</td>
</tr>
</tbody>
</table>

MEETING DATES (IF KNOWN):

STUDENT DECLARATION: I agree to fulfill the above defined contract:

Signature: .................................................. Date: 25/11/2019

SECTION B (Supervisor):

I am willing to supervise and support this project. I have checked the student’s academic record and believe this student can complete the project. I nominate the following examiner, and have obtained their consent to review the report (via signature below or attached email):

Signature: .................................................. Date: 25/12/2019

Examiner: .................................................. Signature: ..................................................
(Nominated examiner(s) may be subject to change on request by the supervisor or course convener)

REQUIRED DEPARTMENT RESOURCES:

SECTION C (Course convener approval)

Signature: .................................................. Date: ..................................................

Research School of Computer Science