

**EVENT DETECTION FOR SMART CONFERENCE ROOM USING SPATIO-  
TEMPORAL CONVOLUTIONAL NEURAL NETWORK**

BY  
TAN YI JIAN

A REPORT  
SUBMITTED TO  
Universiti Tunku Abdul Rahman  
in partial fulfillment of the requirements  
for the degree of  
BACHELOR OF COMPUTER SCIENCE (HONS)  
Faculty of Information and Communication Technology  
(Kampar Campus)

JAN 2020



UNIVERSITI TUNKU ABDUL RAHMAN

**REPORT STATUS DECLARATION FORM**

**Title:** EVENT DETECTION FOR SMART CONFERENCE  
ROOM USING SPATIO-TEMPORAL  
CONVOLUTIONAL NEURAL NETWORK

**Academic Session:** JAN 2020

I TAN YI JIAN  
**(CAPITAL LETTER)**

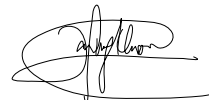
declare that I allow this Final Year Project Report to be kept in  
Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,



(Author's signature)



(Supervisor's signature)

**Address:**  
15, LORONG BUKIT KECIL 6,  
TAMAN BUKIT KECIL,  
14000 BUKIT MERTAJAM,  
PULAU PINANG.

**Date:** 22/04/2020

Tan Hung Khoo  
Supervisor's name

**Date:** 24 April 2020

**EVENT DETECTION FOR SMART CONFERENCE ROOM USING SPATIO-  
TEMPORAL CONVOLUTIONAL NEURAL NETWORK**

BY

TAN YI JIAN

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONS)


Faculty of Information and Communication Technology

(Kampar Campus)

JAN 2020

## DECLARATION OF ORIGINALITY

I declare that this report entitled “**EVENT DETECTION FOR SMART CONFERENCE ROOM USING SPATIO-TEMPORAL CONVOLUTIONAL NEURAL NETWORK**” is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature :  \_\_\_\_\_

Name : TAN YI JIAN

Date : 22/04/2020

## ACKNOWLEDGEMENT

I would like to express my deep appreciation to my project supervisor, Dr. Tan Hung Khoon who has given me the opportunity to engage in this project, **“EVENT DETECTION FOR SMART CONFERENCE ROOM USING SPATIO-TEMPORAL CONVOLUTIONAL NEURAL NETWORK”** and his valuable and constructive guidance and suggestions throughout the development of this project. His willingness to give his time so generously has been very much appreciated.

I would also like to extend my thanks to Dr. Tan Hung Khoon and Ms. Lai Siew Cheng for their help in offering me the computational resources in the development of this project. I would also wish to say thanks to my parents and my family for their love, support and continuous encouragement throughout the course. Finally, I would like to express my appreciation to my project partner, Belinda Khoo Pai Lin for her selfless contribution and collaboration in this project.

## ABSTRACT

Conferencing room is one of the most important workspace in an organization regardless of its organizational domain and scale. Endless of fate changing organizational decisions are made in this workspace. Thus, in order for one to be triumphing over the crowd, one must make certain that the management of the conference room must be the most constructive when compared to their competitors. That being said, there is an increasing trend of organizations attempt to incorporate several of analytic tools in their conference room. The integration of different tools in a conference room is frequently described as a “smart conference room” and being abbreviated as SCR. As there are many different types of analytic tools, this project mainly focuses on the tools that are used to monitor the usage of the conference room.

In the existing SCR systems, the most common techniques used are based on occupancy analysis. Occupancy analysis is a technique aimed to detect the presence of occupants via various sensors such as infrared sensor. However, this technique lack of the capability to model more information about the conference room. In order to overcome this, this project aims to implement the current state-of-the-art human action recognition (HAR) techniques to detect on-going events in a conference room. The HAR technique selected in this project is based on two-stream network with ResNet-34 variant and (2+1)D convolutional blocks. Besides, current state-of-the-art object detection technique which known as You Only Look Once (YOLOv3) will be used for analytical purposes, for instance, counting people in the conference room.

The model will be pretrained on Kinetics dataset and fine-tuned on Conference dataset. The Conference dataset is collected from Company X and will be annotated and pre-processed prior to the training process. Consequently, all the models will be integrated into the web service of SCR system in order to work with other modules in the conference room. Consequently, the system is able to detect on-going events based on the human activities and provide useful analytic insights for effective conference room management.

# TABLE OF CONTENTS

<b>TITLE PAGE</b>	<b>i</b>
<b>DECLARATION OF ORIGINALITY</b>	<b>ii</b>
<b>ACKNOWLEDGEMENT</b>	<b>iii</b>
<b>ABSTRACT</b>	<b>iv</b>
<b>TABLE OF CONTENTS</b>	<b>v</b>
<b>LIST OF FIGURES</b>	<b>vii</b>
<b>LIST OF TABLES</b>	<b>viii</b>
<b>LIST OF ABBREVIATIONS</b>	<b>ix</b>
<b>Chapter 1 Introduction</b>	<b>1</b>
1.1 Project Background	1
1.2 Meeting Event Detection for Smart Conference Room	3
1.3 Project Scope and Objectives	4
1.4 Work Distribution	6
<b>Chapter 2 Literature Review</b>	<b>7</b>
2.1 Smart Conference Room Systems	7
2.1.1 WinLight, a WiFi-based occupancy analysis system	7
2.1.2 Smart Buildings System, an audio based occupancy analysis	9
2.1.3 Summary	11
2.2 Human Action Recognition Techniques	11
2.2.1 Hand-crafted Methods	11
2.2.2 Deep Learning Systems	13
2.2.3 Summary	21
2.3 Object Detection Techniques	21
<b>Chapter 3 Meeting Room Event Detection</b>	<b>24</b>
3.1 System Overview	24
3.2 Event Detection for SCR	25
3.2.1 Targeted Events	25
3.2.2 Proposed Event Detection Method	27
3.3 Dataset Generation and Preparation	30
3.4 Experimental Setup	32
3.5 Experiments	33
3.5.1 RGB Network	33



3.5.2 OF Network	35
3.5.3 Fused Two-Stream Network	36
<b>Chapter 4 Web Application Design</b>	<b>38</b>
<b>Chapter 5 Conclusion</b>	<b>43</b>
<b>BIBLIOGRAPHY</b>	<b>44</b>

## LIST OF FIGURES

<b>Figure Number</b>	<b>Title</b>	<b>Page</b>
Figure 1.1	Examples of Meeting, Non-Meeting and Empty event	4
Figure 2.1:	WiFi-based Occupancy Detector (Zou et al. 2018)	8
Figure 2.2	Meeting Mode Occupancy Estimation Pipeline	9
Figure 2.3	Visualization of Inlier Matches (Wang & Schmid 2017)	12
Figure 2.4	Homography Estimation With and Without Human Detector (Wang & Schmid 2017)	13
Figure 2.5	Multiresolution CNNs Architecture (Karpathy et al. 2014)	14
Figure 2.6	3D CNN architecture (Ji et al. 2013)	15
Figure 2.7	Two-Stream Network Architecture (Simonyan & Zisserman 2014)	16
Figure 2.8	Two-Stream 3D Convolutions (Carreira et al. 2018)	17
Figure 2.9	Inflated Inception-V1 architecture (Carreira et al. 2018)	18
Figure 2.10	Decomposition of 3D Conv into (2+1)D Conv (Tran et al. 2018)	19
Figure 2.11	R3D Network Architecture (Tran et al. 2018)	19
Figure 2.12	Two-Stream Network with Feature Aggregation (Ng et al. 2015)	20
Figure 2.13	Bounding boxes with location prediction and dimension priors (Redmon et al. 2016)	23
Figure 2.14	DarkNet-53 architecture (Redmon et al. 2016)	23
Figure 3.1:	System Framework Overview	24
Figure 3.2	Overview of network architecture	27
Figure 3.3	RGB, Horizontal and Vertical OF frames	28
Figure 3.4	R(2+1)D-34 architecture	29
Figure 3.5	R(2+1)D - 34 layers	29
Figure 3.6	Training accuracy, training loss, learning rate over epochs	33
Figure 3.7	Learning rate over epochs (StepLR/ReduceLRonPlateau)	33
Figure 4.1	Login Page	38
Figure 4.2	Administration Page (Index)	39
Figure 4.3	Monitoring Page for Room A	39
Figure 4.4	Monitoring Page for Room B	40
Figure 4.5	Monitoring Page for Room C	40
Figure 4.6	Data Analytics Page	41
Figure 4.7	Developer Page	42

## LIST OF TABLES

<b>Table Number</b>	<b>Title</b>	<b>Page</b>
Table 1.1	Tasks Distribution	6
Table 3.1	Total training time for different modalities	32
Table 3.2	RGB – Accuracies for experiments	34
Table 3.3	OF – Accuracies for experiments	36
Table 3.4	Final accuracies of RGB, OF and Fused stream	36

## LIST OF ABBREVIATIONS

<i>AP</i>	Access Point
<i>API</i>	Application Programming Interface
<i>CNN</i>	Convolutional Neural Network
<i>FC</i>	Fully-Connected
<i>GMM</i>	Gaussian Mixture Model
<i>GPU</i>	Graphical Processing Unit
<i>HVAC</i>	Heating, ventilation, and air conditioning
<i>LDA</i>	Linear Discriminative Analysis
<i>LSTM</i>	Long Short Term Memory
<i>NLP</i>	Natural Language Processing
<i>OF</i>	Optical Flow
<i>PCA</i>	Principle Component Analysis/
<i>PIR</i>	Passive Infrared Sensor
<i>ReLU</i>	Rectified Linear Unit
<i>RGB</i>	Red Green Blue
<i>S/T</i>	Spatiotemporal
<i>SCR</i>	Smart Conference Room
<i>SGD</i>	Stochastic Gradient Descent
<i>SMR</i>	Smart Meeting Room
<i>SRT</i>	Smart Room Technologies
<i>STIP</i>	Spatiotemporal Interest Point
<i>SIFT</i>	Scale-Invariant Feature Transform
<i>HOG</i>	Histogram Of Oriented Gradients
<i>HOF</i>	Histogram Of Optical Flow
<i>SURF</i>	Speeded Up Robust Features
<i>MFCC</i>	Mel-Frequency Cepstral Coefficients
<i>WinIPS</i>	WiFi-based Non-intrusive Indoor Positioning System
<i>RANSAC</i>	Random Sample Consensus

## CHAPTER 1 INTRODUCTION

### 1.1 Project Background

Conference room is a collaboration workspace that can decide the fate of an organization. Various organizational game changing decisions are made in this workspace. Therefore, creating a smart and efficient conference room can definitely improve the performance of an organization. Unfortunately, most of the companies are still relying on traditional tools and systems to facilitate their daily conference room management tasks. Difficulties in facilitating efficient and engaging meeting can cause a negative impact on organizational performance.

In fact, there are various existing problems in traditional conference room which people are trying to address. Generally, the root of these issues is that the conference room resources in any organization are finite. As a result, workers will need to compete for limited resource in order to use the conference room, especially in peak hour. This can be worsen for most of the international companies because they will need to compromise for geological differences and reserve a specific timeslot for international conference. The specific timeslot will lead to high demand on conference room resources, creating a busy scheduling havoc.

Effective and strategic management of resources plays most important role when the resources are limited. Inefficient management will lead to extremely poor resource utilization. However, resource management is an excruciating pain for most companies because all of the managing tasks are commonly based on traditional system. There are several obvious issues that will faced by traditional system in resource management task. Firstly, most of the traditional conference room reservation scheduling systems are based on predetermined schedules. Since the meeting time span is not always exact as it is scheduled, there are some conditions that a meeting room is underutilized. For example, the meeting ends earlier than its schedule or the cancellation of meeting without releasing the room. Besides, it will also lead to different resource wastage issues such as room hogging and abuse problem. Irresponsible workers may abuse the conference room resource for non-meeting purpose, such as resting and private usage. Moreover, the conference room resource may be wasted by the ineffective system

which reserve the inappropriate size of room for different teams. For instance, small teams are assigned to a large conference room.

Aside from room resources, the system is also playing an important role in regulating energy consumption through control of lighting and Heating, Ventilation, Air Conditioning (HVAC). Ineffective management will lead to energy wastage. For example, the electrical equipment in the room remain unclosed after the meetings ended.

In order to address these issues, various top companies are paying attention and putting effort on designing and building a better conference room. Unlike the past, the computational power has increased to a point that it is just a matter of time that humans can harness the power of it and apply them into the creation of smart conference room (SCR). Indeed, SCR can be developed by a wide range of analytic tools and techniques in order to facilitate the management of conference room. Recently, Microsoft and Google is integrating a variety of analytic tools to enhance the conferencing experience. It is clearly seen that there's an increasing trend of the needs of SCR.

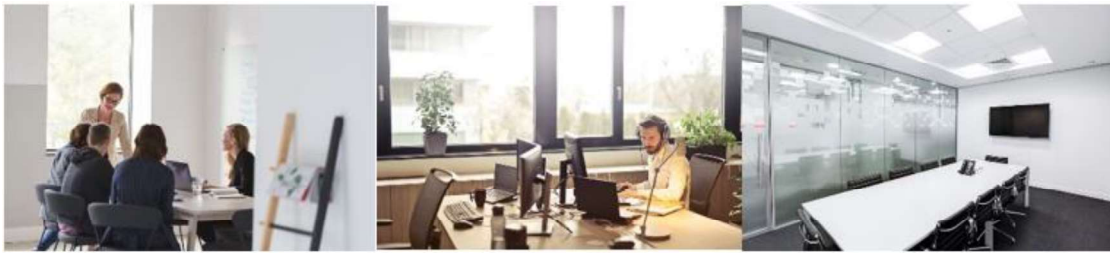
However, majority of the existing systems are merely based on low level analytical techniques such as occupancy analysis to gain the occupancy information on the conference room. Generally, occupancy analysis is a technique which used various sensors to detect the number of the occupants in a conference room. For instances, (Zou et al. 2017) used audio sensors and (Huang et al. 2016) used wireless sensors to detect the occupants in the conference room. The information obtained from occupancy analysis may provide hints for other sub-system such as reservation scheduling system to manage the conference room effectively. However, one of the most significant drawback of these systems is that they are unable to model the conference room information accurately. They are rather restricted to just detect the occupants without any further information. In this case, the system may not be robust enough to manage the conference room in an effective way if they are unable to identify the on-going event in the conference room. If the system is used for room reservation, the reservation system may not be able to work as intended. False prediction of the availability of conference room may lead to ineffective management. For example, in the case of HVAC control, air conditioning will be switched on while there are janitors who are cleaning the conference room. Thus, occupancy analysis approach may not be feasible for this kind of purpose.

The motivation of this project is to propose a way that can resolve the problems stated and improve the features of a SCR system. Unlike the conventional occupancy analysis approaches, this project takes the context from human actions to detect the on-going events in the conference room based on the visual inputs. The underlying technique used is the current state-of-the-art human action recognition. Then, it can be integrated with other systems in SCR. For instance, the proposed system in this project can be integrated with systems such as reservation scheduling system to enable them to function in a more intelligent manner. In addition, the system is able to detect if there's irresponsible workers who try to occupy the room for non-meeting purposes. Besides, this project is based on methods that allow detection in an online and real time. As a consequence, it creates a lot of possibilities for the application of the technique. Especially with the advancement of fourth industrial revolution, the deliverable of this project can be integrated with the internet of things to create a lot of functionalities. For instance, it can be used to identify and recognize the actions of the people who are currently presenting in the conference room while tracking the number of occupants in the conference room. Thus, the system is able to react accordingly via the connected devices in the conference room.

## **1.2 Meeting Event Detection for Smart Conference Room**

This project proposed a SCR module which is able to accurately detect the meeting and non-meeting events in the conference room via the context of human action. In fact, human action recognition is a recent state-of-the-art technique in video analysis task. Unlike still image classification, video analysis task is required to take account of the temporal information in addition to the spatial information. In this way, the system is able to predict the presented human action based on the appearance and motion information from the visual input. Then, the system predicts the on-going event in the conference room based on the action information. In fact, system is able to find the clues based on some of the common activities in the meeting event. For example, meeting events typically consist of clues like the body language of the speakers. In contrary, the system will also learn the existing clues from the action where janitors are cleaning the room, repair man is trying to maintain the room or even an irresponsible worker who is trying to take a nap in the conference room. Action recognition task attempts to learn all the possible actions which are likely to happen in meeting or non-

meeting events. Other than meeting and non-meeting events, the system will also learn the background class, which is the situation where the room is empty.



**Figure 1.1 Examples of Meeting, Non-Meeting and Empty event**

In order to create a SCR module which is able to provide useful analytics, it is required to be capable to detect the presence of occupant in the conference room, meanwhile, detecting the on-going actions and events in the room. Based on these information, there are different types of analytics which can be generated. Firstly, the usage rate of the room can be analyzed in order to devise the conference room policies and enhance the scheduling systems of the conference rooms. This can be achieved by gaining insight from the analytics to identify the demands of conference room in different timeslots and even different department. For instance, based on these insights, companies can deduce if the expansion of meeting rooms will help their organizational performance. This is especially true when different departments in a company have different pattern of meeting, hence, they have different demands on the conference room resources.

Besides, other system is also able to rely on the prediction to schedule the room reservation such that no workers are able to abuse the room for inappropriate usage. Moreover, the system is also capable to control the HVAC based on the on-going events in the meeting room. For instance, the system may only turn on the light when janitors are cleaning the room instead of turning on the air-conditioning for just mere 5 minutes cleaning routine.

### **1.3 Project Scope and Objectives**

This project aims to build a real-time web-based system that is able to detect the on-going events in the conference room and provide insightful analytics. Unlike the conventional occupancy analysis approaches, this system classifies if there are meeting activities or non-meeting activities in the conference room based on the human



activities identified. Besides, it is also integrated with analytic tools as an extension to showcase the capabilities of the system.

The objectives of this project are as below:

**A deep learning system for detecting meeting events in conference videos.**

The main impact of this project is to build a novel system that is able to perform human action recognition in conferencing environments in real time and online fashion. The proposed SCR module distinguishes if there is an ongoing meeting activity or event based on the action context. Specifically, the events are meeting, non-meeting and empty. With the combination of high detection speed and high accuracy of selected framework, the system is able to work effectively with other systems in the smart conference room.

**A new dataset for training meeting event detection.**

The most crucial building block of this project is to generate a large dataset, which are the footages in conference room. The footages of conferencing activities in an organization are normally private and confidential. The organizations are seldom willing to provide their resources since the footages contain sensitive content. Hence, self-generating datasets from scratch is inevitable in this situation. Generation of datasets includes collecting, cleaning and annotating the footages. Other than manually generating the footages, task of labeling the footages can be taxing as well. This is due to the reason that given a single footage, it will contain multiple action instances that can start and end at arbitrary time frame. In order to label the footages correctly, all of them need to be gone through entirely and labeled manually.

**Data analytics for smart conference room.**

The proposed system also produces meeting analytics that provide insights for effective conference room management. For instance, the system is able to track the number of occupants and available seats in the conference room. Occupancy rates information of conference rooms are useful because the information may be used to deduce if the conference room is abused or underused. Therefore, the reservation scheduling system can utilize this information to effectively manage the conference room, for example, schedule a smaller conference room for smaller group of occupants.

### 1.4 Work Distribution

This is an industrial project and is done in the collaboration with a multi-national company. This joint project is done in a group of 2 persons whereby the other member is Belinda Khoo Pai Lin. The works and tasks of this project are distributed equally. The allocated tasks are as follow.

Data preparations including cleaning, annotation and preprocessing ( $\frac{1}{2}$ of the dataset, ~100 hours)	Belinda, Yi Jian
Implementation of RGB stream in Two-stream network	Yi Jian
Implementation of OF stream in Two-stream network	Belinda
Fusion of Two-stream network	Belinda, Yi Jian
Flask web application for administrator mode	Belinda
Flask web application for developer mode	Yi Jian
Implementation of YOLOv3 for object counting	Belinda, Yi Jian

**Table 1.1 Tasks Distribution**

## **CHAPTER 2 LITERATURE REVIEW**

This project exploits human action recognition and object detection technique to build a smart conference room. Therefore, this chapter is divided into 3 sections which are 2.1, 2.2 and 2.3 which review the existing works on smart conference room system, human action recognition technique and object detection technique respectively.

### **2.1 Smart Conference Room Systems**

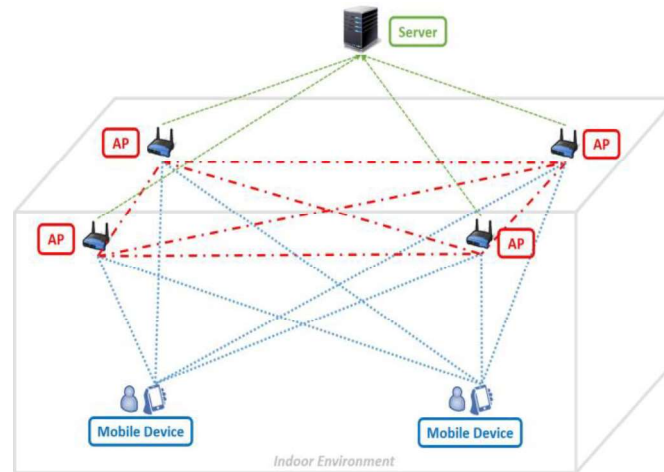
This section analyzes and discusses about the current existing smart conference room systems. In fact, there are various types of SCR systems which are built for different purposes. Each of them have different purposes for aiding companies in managing conference activities. Some of them are specialized to improve the scheduling task for room reservation while the others automate different parts of the conference activities.

However, most of the existing SCR systems mainly focuses on exploiting occupancy analysis technique. They aim to analyze the occupancy condition in the conference rooms and automate some process based on the occupancy information. For instance, HVAC system of the room can be controlled by detecting the presence of occupant.

They perform occupancy analysis and estimation based on the collected sensors' data such as temperature, audio and passive infrared (PIR) and WiFi.

#### **2.1.1 WinLight, a WiFi-based occupancy analysis system**

WinLight is a lighting control system which is based on WiFi for occupancy analysis. The proposed work use occupancy analysis approach to detect if there's any occupant(s) inside the room. They achieved this by using 'WiFi-based non-intrusive indoor positioning system', the authors also abbreviated it as 'WinIPS'. The system design is as shown in Figure 2.1.



**Figure 2.1: WiFi-based Occupancy Detector (Zou et al. 2018)**

WinIPS is designed to extract the occupants' information based on their mobile devices. Several WiFi access points (AP) are required to be installed in place to expand the coverage of WiFi signal to the entire room. The access points installed will then capture the data packets in the network to obtain the information of occupants. The underlying technique of this approach is based on the active AP scan of the client devices. Specifically, the connected clients' radio will constantly transmits probe requests to the AP and wait for the AP to response. The probe requests from the client devices will contain useful information to be extracted which is the WiFi signal strength. The signal strength is determined by the WiFi connection strength and it is typically affected by distance between the client devices and the AP. The information will then be extracted by the backend server. If the occupants reach the threshold for the weak signal strength, which is set to be  $-95\text{dBm}$ , the system will identify the condition as the occupants are leaving the room.

The author proposed the system to be used to control the lighting of smart conference room depending on the occupancy condition of the conference room. If there's any people inside the room, the backend server will detect them by performing threshold on their devices' WiFi signal strength, it will then switch on the lights in the room.

The major strength of this system is that it can work in non-intrusive manner. It does not require manual user intervention or input in order to identify the location of the occupants. Eliminating the need for user intervention by automating manual tasks is the crucial block of building successful SCR system.

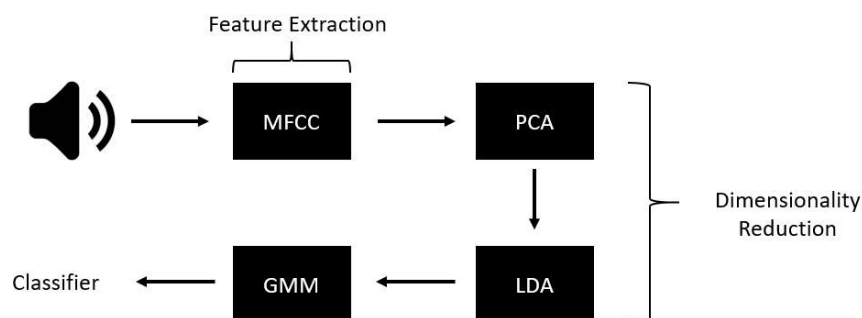
However, it is based on the assumption that the users are carrying their handheld devices throughout the day. It also requires users to install dedicated application in their mobile devices. Else, the system will not detect them. This make the system to be impractical. Apart from that, it does not actually take the context of occupants' activities into account. It only cares about the presence of occupants without considering their activities in the conference room. It is not able to react appropriately based on the occupants' activities. For example, it will switch on the air-conditioning even there's only a janitor who is running daily cleaning routine.

In this project, the system is required to detect on-going events in the conference room. WiFi-based occupancy analysis method is not feasible for this project.

### 2.1.2 Smart Buildings System, an audio based occupancy analysis

Unlike the previous work (WiFi-based occupancy analysis), (Huang et al. 2016) proposed an approach that exploits audio and speech processing to perform occupancy analysis. They proposed to classify the room occupancy estimation problem into 2 scenarios which are meeting mode and party mode. Meeting mode is the scenario where each of the speakers speak separately, and their voices are non-overlapping. On the other hand, party mode refers to the scenario where multiple people speak at the same time. Different models are devised to handle the two modes.

**Meeting Mode.** In this mode, only one person speaks at any one time. Since the voice of the speakers are non-overlapping, the speakers are by definition separable. Hence, the voice of the speakers in the room are able to be modeled using the Gaussian Mixture model. The voice classification pipeline is as shown below.



**Figure 2.2 Meeting Mode Occupancy Estimation Pipeline**

The system utilize a common speech feature known as Mel-Frequency Cepstral Coefficients (MFCCs) to convert audio frames into high dimensional feature vector. Principle Component Analysis (PCA) and Linear Discriminative Analysis (LDA) are then used to reduce the dimension of the features in addition to maximize the separation of different class. As a result, the features that are connected with speakers are used to build a Gaussian Mixture Model (GMM), which will represent the characteristic of the specific speaker's voice. Later, based on the trained GMM classifier, the one where its PCA/LDA features are most likely and closely related to be the hypothesis speaker will be selected. Therefore, the occupancy rate can be deduced using the number of identified speakers.

**Party Mode.** In this mode, multiple participants may talk at the same time. Hence, the model needs to be able to distinguish the voice signatures from different number of people. This is achieved by applying Short-Time Energy (STE) in occupancy estimation. It will be divided into 2 steps. Firstly, random crowd speech will be generated with decay modelling. Estimation will be made on the STE level associated with different sizes of crowds. Eventually, the accuracy will be tested. Unfortunately, the STE estimation may not work effectively when there are 10 to 20 people in the room, as tested in the original implementation by (Huang et al 2016). They believed that the low accuracies in 10 to 20 speakers are caused by the STE energy levels are similar for both 10 and 20 speakers. This technique will be infeasible for smart conference room application since the number of occupants will likely fall between 10 and 20.

The strength of this work is similar to the previous one that is non-intrusive. Besides, it is also cost effective. Only microphones and a backend server are needed to being set up. However, one of the techniques used in this work performs inefficiently when the size of occupants is between 10 and 20 people despite the fact that it can be improved over time. Lastly, and most importantly, the audio processing way of occupancy estimation is unable to differentiate between meeting activities and non-meeting activities. For instance, the system is incapable of distinguishing a meeting discussion from normal chitter-chatter between people. This approach is only able to predict the number of occupants in the conference room. Thus, it is infeasible for this project as it fails to detect the on-going event in the conference room.

### **2.1.3 Summary**

In summary, most of the existing SCR systems mainly focuses on occupancy analysis to detect the occupants in the conference room. Occupancy analysis approach will only tell if there's occupant inside the conference room instead of telling what is happening in the conference room. They lack of the capability to identify the on-going event in the meeting room. In fact, the information provided by occupancy analysis is not sufficient to make other SCR systems work efficiently.

In this project, the system is required to take the context of human activities into account. By including the human action information into the system, the system is able to predict the on-going event in the conference room.

## **2.2 Human Action Recognition Techniques**

Action recognition is a computer vision task which tackle on video analysis. Action recognition task involves the recognition of actions from videos where there may or may not be an action in the video. This section will discuss in depth about the related works in human action recognition task.

### **2.2.1 Hand-crafted Methods**

Prior to the deep learning era, there are a lot of works which are fundamentally based on shallow learning on hand-crafted spatio-temporal features. For instance, spatio temporal interest points (STIP), Scale-Invariant Feature Transform (SIFT), Histogram of Optical Flow (HOF) and Histogram of Oriented Gradients (HOG) are some of the most popular video representation approach in the early 2000s. Thenceforth, (Scovanner et al. 2007) formulated a new variant of 3D-SIFT descriptor which was able to outperform these previous descriptors. However, the state-of-the-art result is quickly outperformed by HOG3D by (Klaser et al. 2008). Later, (Laptev et al. 2009) had proposed to utilize HOG3D together with dense spatio-temporal interest point detector for performing action classification and the detector is proven to be able to outperform the other approaches.

Besides, there are some shallow video representation approaches which adapt the local descriptor to support region. For instance, (Wang et al. 2011) proposed an approach to

take advantage of the optical flow algorithm to compute dense point trajectories. Later, it was improved by (Wang & Schmid 2017) and known as improved Dense Trajectories (iDT). (Wang & Schmid 2017) proposed to explicitly estimate camera motion in order to improve dense trajectories. In order to estimate the camera motion, they assume that the 2 successive frames are interrelated by a homography. Hence, they estimate the homography between frames by discovering the correspondences between them. This is achieved by complementing 2 different approaches as shown:

1. They match the extracted Speeded Up Robust Feature (SURF) from frames using nearest neighbor rule.
2. Then, they also select motion vectors from optical flow for salient feature points by using good-features-to-track criterion.

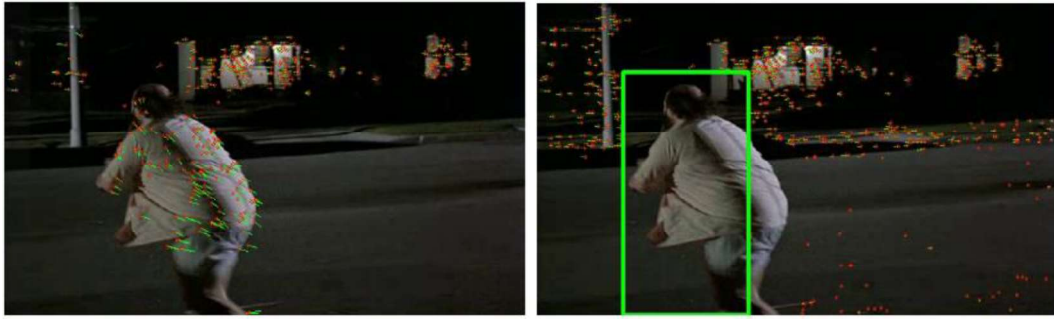
By doing this, (1) will extract blob-type features while (2) focuses on corners and edges. Then, they estimate the homography with random sample consensus (RANSAC) using the matched points from both approaches. The inlier matches between 2 consecutive frames of the estimated homography can be visualized in Figure 2.3, where green arrows represent the SURF matches and red arrows represent the dense optical flow.



**Figure 2.3 Visualization of Inlier Matches (Wang & Schmid 2017)**

However, the inlier matches are not ideal as they also includes human motions. To address this issue, they proposed to remove the human matches by truncating the matches in the human regions. They use the part-based human detector which is proposed by (Prest et al. 2012). The part-based detector will detect the bounding boxes which contain human body parts. The bounding boxes are used as masks to prune the feature matches when estimating homography. The result is as shown below.





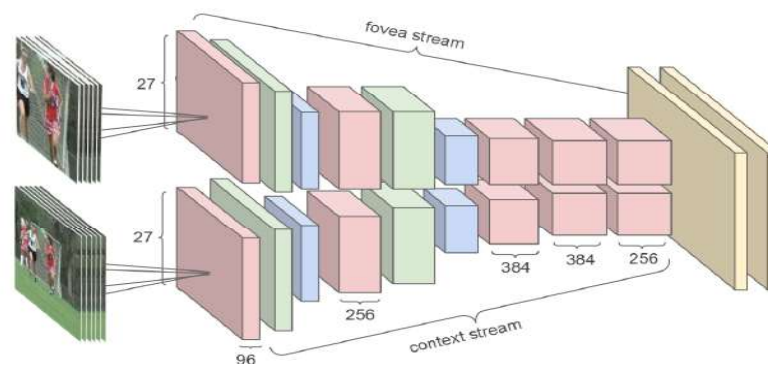
**Figure 2.4 Homography Estimation With and Without Human Detector (Wang & Schmid 2017)**

As a result, they proved that the performance of dense trajectories can be improved by warping optical flow and removing background trajectories with estimated homography. However, this approach relies heavily on hand crafted features and the performance is susceptible to be generalize well on different tasks.

### 2.2.2 Deep Learning Systems

After the breakthrough of CNNs, there are various researchers attempt to build deep architecture for action recognition. Generally, the communities are based on 2 main approaches which are 2D and 3D convolutional operations to perform action recognition. Then, based on the 2D and 3D convolutions, they attempted various design to combine the spatial and temporal information.

**2D convolution approaches.** Due to the massive achievement of 2D convolution in image classification domain, there are several research tried to reuse the idea of 2D convolutions into video action recognition domain. In 2D Conv, 2D feature maps are applied with convolutions to extract features from the 2D spatial dimensions. In this way, the video inputs are treated as separated static frames, resulting the design similar to standard design in image classification domain. Notably, (Karpathy et al. 2014) proposed a novel approach which use multiresolution 2D CNNs to handle the feature extractions from each frame in the stack independently. The design is as shown in Figure 2.5.



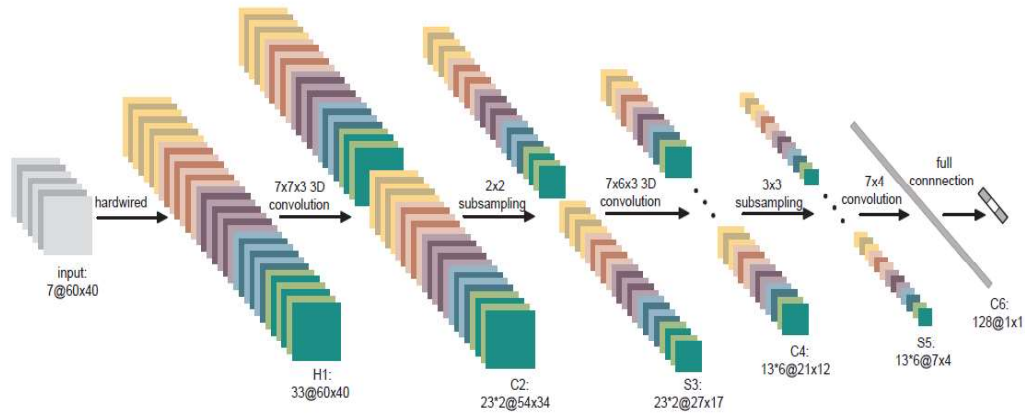
**Figure 2.5 Multiresolution CNNs Architecture (Karpathy et al. 2014)**

The architecture proposed by (Karpathy et al. 2014) consists of 2 separated stream, each of them will handle different resolution inputs. Given the input of size  $178 \times 178$ , the first stream (context) takes downsampled frames (halved size of  $89 \times 89$ ) from the inputs. On the other hand, the second stream (fovea) takes the center region at the original resolution ( $89 \times 89$ ). Hence, the input dimensionality is halved in this design. Eventually, the both streams' activation will be concatenated together and being fed into FC layer with dense connections. Then, the predictions across the frames will be pooled by either using different fusion technique.

In the implementation proposed by (Karpathy et al. 2014), the multiresolution design is able to significantly reduce the computational time by a factor of  $2 \sim 4$  while retaining the comparable classification accuracy to standard 2D convolution. However, this approach completely ignores the temporal structure of the video. For instance, the model is unable to differentiate between the action of opening and closing a door. In contrary, 3D convolution based approaches is able to model the temporal information better from video input when compared to 2D based approaches.

**3D convolution approaches.** 3D convolution is another widely used method in the video domain. Indeed, 3D seems like a more natural approaches to deal with video classification than 2D convolutions. This is due to the nature that video inputs are considered to be 3 dimensional inputs which incorporate the temporal dimension in addition to spatial dimension. Hence, 3D convolution can be seen as another extension of 2D convolution but with spatio-temporal (3D) filters. Notably, (Ji et al. 2013) are the first to propose to use 3D CNN architecture in action recognition task. They proposed to use 3D filter to convolve over the cube (shaped by stacking of frames). As a

consequence, the feature maps is able to capture motion information because it is connected to several continuous frames in the former layer. The proposed architecture is as shown below.



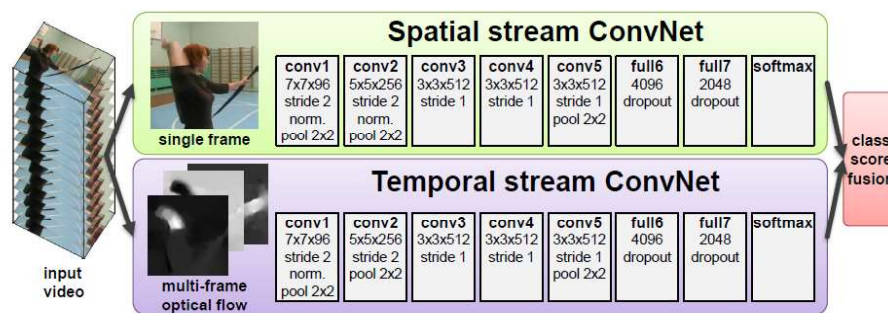
**Figure 2.6 3D CNN architecture (Ji et al. 2013)**

In their proposed architecture, they use 7 frames of size  $60 \times 40$  as inputs to their 3D CNN model. They apply hardwired kernels to produce 5 channels of information from the inputs which consist of gray pixel values, gradients and optical flow computed in horizontal and vertical direction. They believed that this hardwired scheme which based on prior knowledge on features is superior to random initialization. Then, they applies 3D convolutions on the layers as shown in Figure 2.6. Note that they perform the 3D convolution separately on each of the 5 channels. They also apply 2 sets of different convolutions at each location to double the feature maps as shown in C2 and C5. Besides, they also apply subsampling to reduce the spatial dimension. As a conclusion, 7 consecutive frames are converted into a 128 dimensional feature vector which is capable to represent the motion information across the 7 frames. The final classification is obtain by feeding the feature vector into linear classifier.

As a result, 3D convolutions approaches are shown to be able to outperform some of the 2D based approaches due to the better design in capturing the motion information across the frames. However, this method only designed to be operating on hand-wired combination of multiple input channels. Besides, 3D convolutions require much higher computational cost when compared to 2D convolutions due to the reason that 3D convolutions contain significantly more parameters to be optimized. Moreover, 3D

convolutions are also relatively more prone to overfit, resulting in harder training and optimization process.

**Two-stream approaches.** Other than plain 2D and 3D based design, there are several works which proposed different ways to incorporate the temporal information into the design of the network. Notably, (Simonyan & Zisserman 2014) proposed the architectures of two stream networks to fuse spatial information with the temporal information. In their implementation, the spatial information is represented by a stream of static frames while the temporal information is represented by a stream of optical flow frames. Optical flow is another widely used approach to encode motion patterns by calculating the motion difference between 2 consecutive frames. The architecture design of their work is as shown below.



**Figure 2.7 Two-Stream Network Architecture (Simonyan & Zisserman 2014)**

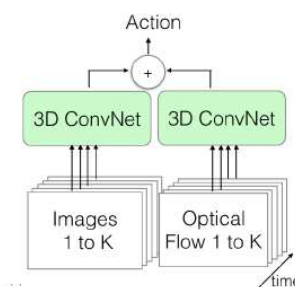
In their design, the spatial stream network takes RGB frame as its input while temporal network stream takes a stack of optical flow frames. Both streams will be fed into separated but identical network. Then, the scores from both stream will be obtained by applying softmax layer. In order to incorporate the temporal structure into the network, they proposed to fuse multiple modalities (RGB and optical flow) by late fusion. This is achieved by averaging the scores of both streams at the final classification layer. By doing so, their proposed method are able to outperform the 3D convolution based approaches while still retaining relatively low computational time.

However, this approach is not able to model the pixel-wise correspondences between spatial and temporal features as the fusion is done on the final classification scores only. Besides, it has limited temporal scales since the spatial stream only operates on single

RGB frame and the temporal stream only operates on fixed number of optical flow frames.

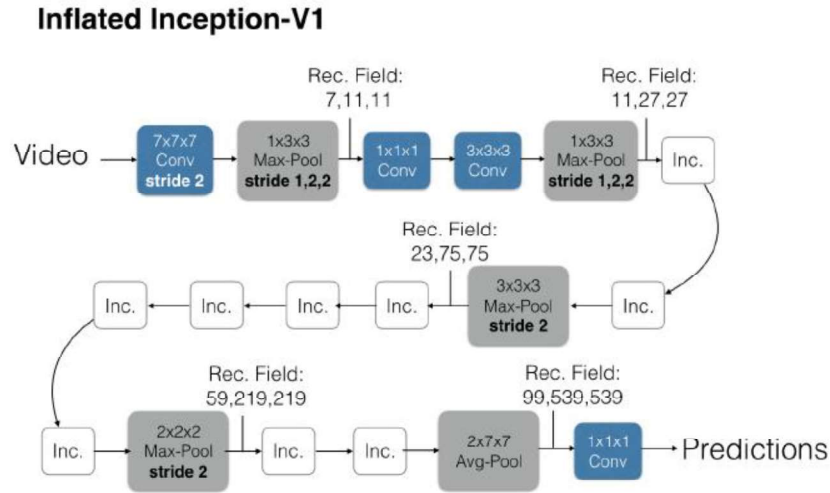
**Modality fusion.** In order to solve the drawbacks presented in two-stream network proposed by (Simonyan & Zisserman 2014), (Feichtenhofer et al. 2016) conducted an empirical study on different spatial, temporal and modality fusion techniques and proposed a novel spatio-temporal fusion architecture for two-stream network. Unlike original two-stream network, they proposed to fuse the stream at the last convolutional layer with their activation maps instead of fusing at the FC layer. By doing so, the spatial correspondences of the appearance and motion are retained. Their proposed work has slightly improved performance when compared to the original implementation of two-stream network.

**Two-stream 3D.** (Carreira et al. 2018) further extend the previous work by proposing a novel architecture known as Two-Stream Inflated 3D ConvNets (I3D). I3D is built based on the state-of-the-art image classification architecture, but with inflated 3D filter and pooling kernels. In their implementation, they inflate the 2D square kernel  $N \times N$  into 3D cubic kernel  $N \times N \times N$  and bootstrap the parameters from pretrained ImageNet model. Bootstrapping is done by duplicating N times for the weights of 2D filters and rescaling them by dividing N. They proposed to perform 3D convolution on both stream of inputs (RGB and OF) as shown below.



**Figure 2.8 Two-Stream 3D Convolutions (Carreira et al. 2018)**

Note that they also use inflated Inception-v1 as the backbone of their network architecture. However, they do not perform temporal pooling in the first and second max pooling layers. Symmetric kernels and stride are used in the rest max-pooling layers.  $2 \times 7 \times 7$  kernel is used for the final average pooling layer. The architecture is as shown in figure below.



**Figure 2.9 Inflated Inception-V1 architecture (Carreira et al. 2018)**

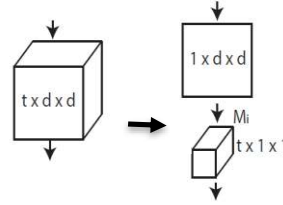
By this implementation, I3D is able to outperform other approaches on HMDB51 and UCF101 dataset based on the pretrained model on ImageNet. However, I3D still share the drawbacks of common 3D based approaches. Despite the state-of-the-art accuracy, I3D demands significantly larger computational resources than other approaches.

**(2+1)D convolution approaches.** In order to reduce the computational time, (Tran et al. 2018) proposed to factorize the 3D convolutions into discrete spatial and temporal components. Their studies led to the creation of a novel convolutional block “(2+1)D”. Indeed, (2+1)D can be viewed as the middle ground of full 3D convolution and 2D convolution. The novel architecture proposed by (Tran et al. 2018) is known as R(2+1)D as they implement (2+1)D convolution using residual blocks to address the vanishing gradient issue. In their work, they empirically demonstrated the performance differences between various convolutions such as 2D, 3D and mixed convolutions (MC) in the ResNet framework.

In order to compare (2+1)D convolution with other convolutions in detail, let the input  $x$  to be a 4D tensor of shape  $L \times 3 \times H \times W$ , where  $L$  denotes the number of frames in the stack,  $H$  and  $W$  denotes the spatial dimension and 3 denotes to the RGB channels. The following paragraphs will contrast different convolutional technique in details.

As discussed before, 2D convolution operation completely collapsed temporal information of the video and treat  $L$  value as the number of channels, resulting in 3D input tensor of size  $3L \times H \times W$ . After the convolution with  $N_{i-1} \times d \times d$  spatial filter

( $d$  denotes spatial dimension), the output tensor of  $i$ -th residual block can be represented as  $N_i \times H_i \times W_i$ , where  $N_i$  represents the number of convolutional filters applied on  $i$ -th block. In contrary, the 3D convolution with spatio-temporal filter  $N_{i-1} \times t \times d \times d$  will result in the output of size  $N_i \times L \times H_i \times W_i$ , showing that it does retain the temporal information in 4D video input. Note that  $t$  denotes temporal extent of the filter.



**Figure 2.10 Decomposition of 3D Conv into (2+1)D Conv (Tran et al. 2018)**

Similar to 3D convolutions, (2+1)D takes 4D inputs without collapsing, but carry out convolution with the decomposed version of convolutional filters. (Tran et al. 2018) proposed to factorize  $N_i$  3D convolutional filters of size  $N_{i-1} \times t \times d \times d$  into  $M_i$  2D convolutional filters of size  $N_{i-1} \times 1 \times d \times d$  and  $N_i$  temporal convolutional filters of size  $M_i \times t \times 1 \times 1$  as shown in Figure 2.10. In order to match the number of parameters in (2+1)D block with full 3D block, they formulate the  $M_i$  to be:

$$M_i = \left\lfloor \frac{td^2N_{i-1}N_i}{d^2N_{i-1}+tN_i} \right\rfloor \quad (1)$$

(Tran et al. 2018) used the 18 and 34 layers variant of ResNet architecture for their empirical studies as shown in Figure 2.11. Note that the R(2+1)D architecture is exactly the same as R3D.

layer name	output size	R3D-18	R3D-34
conv1	$L \times 56 \times 56$	$3 \times 7 \times 7, 64, \text{stride } 1 \times 2 \times 2$	
conv2_x	$L \times 56 \times 56$	$\begin{bmatrix} 3 \times 3 \times 3, 64 \\ 3 \times 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3 \times 3, 64 \\ 3 \times 3 \times 3, 64 \end{bmatrix} \times 3$
conv3_x	$\frac{L}{2} \times 28 \times 28$	$\begin{bmatrix} 3 \times 3 \times 3, 128 \\ 3 \times 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3 \times 3, 128 \\ 3 \times 3 \times 3, 128 \end{bmatrix} \times 4$
conv4_x	$\frac{L}{4} \times 14 \times 14$	$\begin{bmatrix} 3 \times 3 \times 3, 256 \\ 3 \times 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3 \times 3, 256 \\ 3 \times 3 \times 3, 256 \end{bmatrix} \times 6$
conv5_x	$\frac{L}{8} \times 7 \times 7$	$\begin{bmatrix} 3 \times 3 \times 3, 512 \\ 3 \times 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3 \times 3, 512 \\ 3 \times 3 \times 3, 512 \end{bmatrix} \times 3$
	$1 \times 1 \times 1$	spatiotemporal pooling, fc layer with softmax	

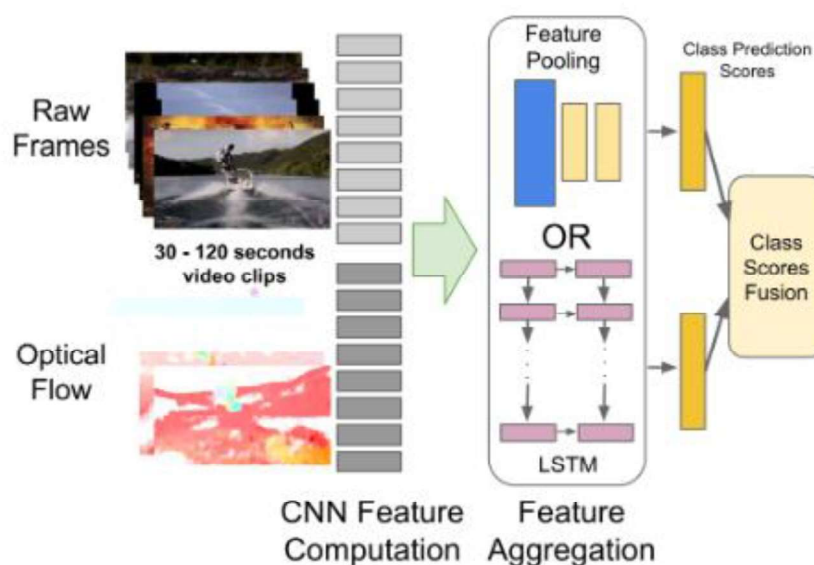
**Figure 2.11 R3D Network Architecture (Tran et al. 2018)**

In order to implement the downsampling operation, there are 1 spatial striding of  $1 \times 2 \times 2$  at conv1 and 3 spatiotemporal striding of  $2 \times 2 \times 2$  at the first layer of conv3, conv4 and conv5. The output tensor size of the last convolutional layer of R(2+1)D is  $\frac{L}{8} \times 7 \times 7$ .

As a result, the model proposed by (Tran et al. 2018) is capable of representing more complex functions due to its increased non-linearities. Other than that, it has lower training and testing loss than full 3D approaches. This is due to the reason that (2+1)D blocks are easier to be optimized when compared to full 3D blocks.

However, this approach still poses the issue where the temporal modelling is still restricted to fix-sized frames per stack (value of L). This means that only part of the full action information is modelled.

**Feature Aggregation.** Other than performing fusion on the final scores directly (as shown in original R(2+1)D implementation), feature aggregation is specifically designed to aggregate the features before feeding into the final classification layer. In the earlier time, (Ng et al. 2015) proposed 2 different approaches to aggregate the outputs from each frames into video level predictions. The first approach is to use max pooling to aggregate local features through time. On the other hand, the second approach use LSTM network to take in the sequences of CNN activations as inputs. Both approaches are shown in Figure 2.12.



**Figure 2.12 Two-Stream Network with Feature Aggregation (Ng et al. 2015)**



In the implementation proposed by (Ng et al. 2015), they used CNN to extract the features from static RGB and optical flow frames. Then, both stream of extracted features will then being fed into either feature pooling layer or LSTM network for feature aggregation. Both scores from each stream will then being fused to get the final output. They proved that the idea of incorporating complete information across the frames are significant for better video classification performance.

In fact, this shows the possible improvement which can be made to resolve the temporal restriction presented in the implementation of R(2+1)D. By aggregating the features before the final classification, the R(2+1)D is no longer restricted to fixed size of inputs.

### 2.2.3 Summary

Based on the reviews on various techniques and architectures, the best performer in terms of detection accuracy and speed is the work by (Tran et al. 2018). This project favors the approach with higher detection speed as it is one of the system requirement. Event detection will only be feasible with performance close to real-time detection. In this case, R(2+1)D is more practical as it can perform prediction in real time manner. However, based on the inspiration of work by (Ng et al. 2015), the temporal modelling limitation in R(2+1)D is possible to be addressed by using LSTM to perform temporal fusion. However, due to the computational resource restriction, the temporal fusion technique will be implemented in the future work.

## 2.3 Object Detection Techniques

Object detection is another crucial computer vision task which classify and detect the location of visual objects in given image. In this project, object detection technique is used as an extension to detect the occupants and seats in the conference room. This section will discuss about the related work in object detection domain. Generally, object detection can be categorized into “two-stage detection” and “one-stage detection”.

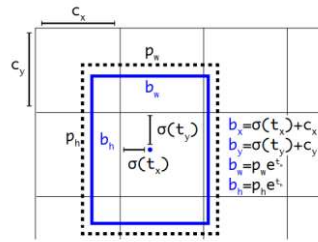
**Two-stage detection methods.** There are some popular two-stage approaches such as Region-based ConvNet (RCNN), Fast RCNN and Faster RCNN. Two-stage methods typically comprise of separated region proposal step and classification step.

Firstly, (Girshick et al. 2014) proposed the RCNN approach which use Selective Search method to extract a set of object proposal (2000 proposals). The extracted region proposals are then fed into ConvNet to obtain a 4096-dimension feature vector and followed by a linear SVM for final classification. This drawback of this approach is that it is computationally inefficient since it requires to perform feature extraction on every region candidates.

Thenceforth, (Girshick. et al 2015) addressed this by proposing Fast RCNN. This approach directly input the image into the CNN to obtain convolutional feature map instead of finding the region proposals first. Besides, they proposed to use region-of-interest pooling (RoIPooling) to wrap and resize the proposed region into fixed size square. As a result, Fast RCNN is able to outperform RCNN by both detection accuracy and speed. Despite the significant improvement of Fast RCNN over vanilla RCNN, the detection speed is still bottlenecked by the proposal detection stage.

Later, (Ren et al. 2015) proposed a technique which known as Faster RCNN to eliminate the bottleneck of proposal detection stage in RCNN and Fast RCNN by substituting the Selective Search algorithm by Region Proposal Network (RPN). In addition, most of the individual blocks of object detection pipeline have been integrated into an end-to-end learning framework. As a result, Faster RCNN outperformed the other approaches.

**One-stage detection methods.** One-stage detectors are generally faster in term of detection speed due to its combined region proposal and classification step. One stage detectors attempt to feed the full image into a single network instead of performing separate proposal detection and verification. (Redmon et al. 2016) was the first to propose “one-stage detection” method which is known as You Only Look Once (YOLO) method. This approach is discussed with the reference to the latest version at the moment, which is YOLOv3. YOLO splits the image into  $S \times S$  grid cell. The object will be detected by the particular grid cell where the center of the object is fallen into. YOLO will retrieve 3 bounding boxes from each grid cell based on the implementation of anchor boxes. The network will then classify each bounding box to obtain its objectness score and offset values (bx, by, bw, bh) as shown in Figure 2.13.



**Figure 2.13 Bounding boxes with location prediction and dimension priors (Redmon et al. 2016)**

$b_x, b_y, b_w, b_h$  represent the predicted (x, y) coordinates, width and height respectively.  $t_x, t_y, t_w, t_h$  are the outputs of the network.  $p_w, p_h$  are the dimensions of the anchor for the box.

In fact, the output of the network is actually a feature map with  $(B \times (5 + C))$  entries, where 5 includes the objectness score and the 4 offset values (bx, by, bw, bh), C represent class confidences, B represent the number of bounding boxes. In addition, they propose to make prediction at 3 scales.  $1 \times 1$  kernel is applied on the feature maps of three sizes (having strides of 32, 16 and 8 respectively) to acquire the prediction. As a consequence, YOLO used 9 anchor boxes as there are 3 different scales. Threshold will also be defined in order to filter out the bounding boxes with lower confidence score. YOLO implemented feature extraction by using DarkNet-53 network. The output of the extractor will be a tensor that contains objectness score and offset values. The network architecture is as shown in Figure 2.14.

Type	Filters	Size	Output
Convolutional	32	$3 \times 3$	$256 \times 256$
Convolutional	64	$3 \times 3 / 2$	$128 \times 128$
Convolutional	32	$1 \times 1$	$128 \times 128$
Convolutional	64	$3 \times 3$	
Residual			$128 \times 128$
Convolutional	128	$3 \times 3 / 2$	$64 \times 64$
Convolutional	64	$1 \times 1$	$64 \times 64$
Convolutional	128	$3 \times 3$	
Residual			$64 \times 64$
Convolutional	256	$3 \times 3 / 2$	$32 \times 32$
Convolutional	128	$1 \times 1$	$32 \times 32$
Convolutional	256	$3 \times 3$	
Residual			$32 \times 32$
Convolutional	512	$3 \times 3 / 2$	$16 \times 16$
Convolutional	256	$1 \times 1$	$16 \times 16$
Convolutional	512	$3 \times 3$	
Residual			$16 \times 16$
Convolutional	1024	$3 \times 3 / 2$	$8 \times 8$
Convolutional	512	$1 \times 1$	$8 \times 8$
Convolutional	1024	$3 \times 3$	
Residual			$8 \times 8$
Avgpool		Global	
Connected		1000	
Softmax			

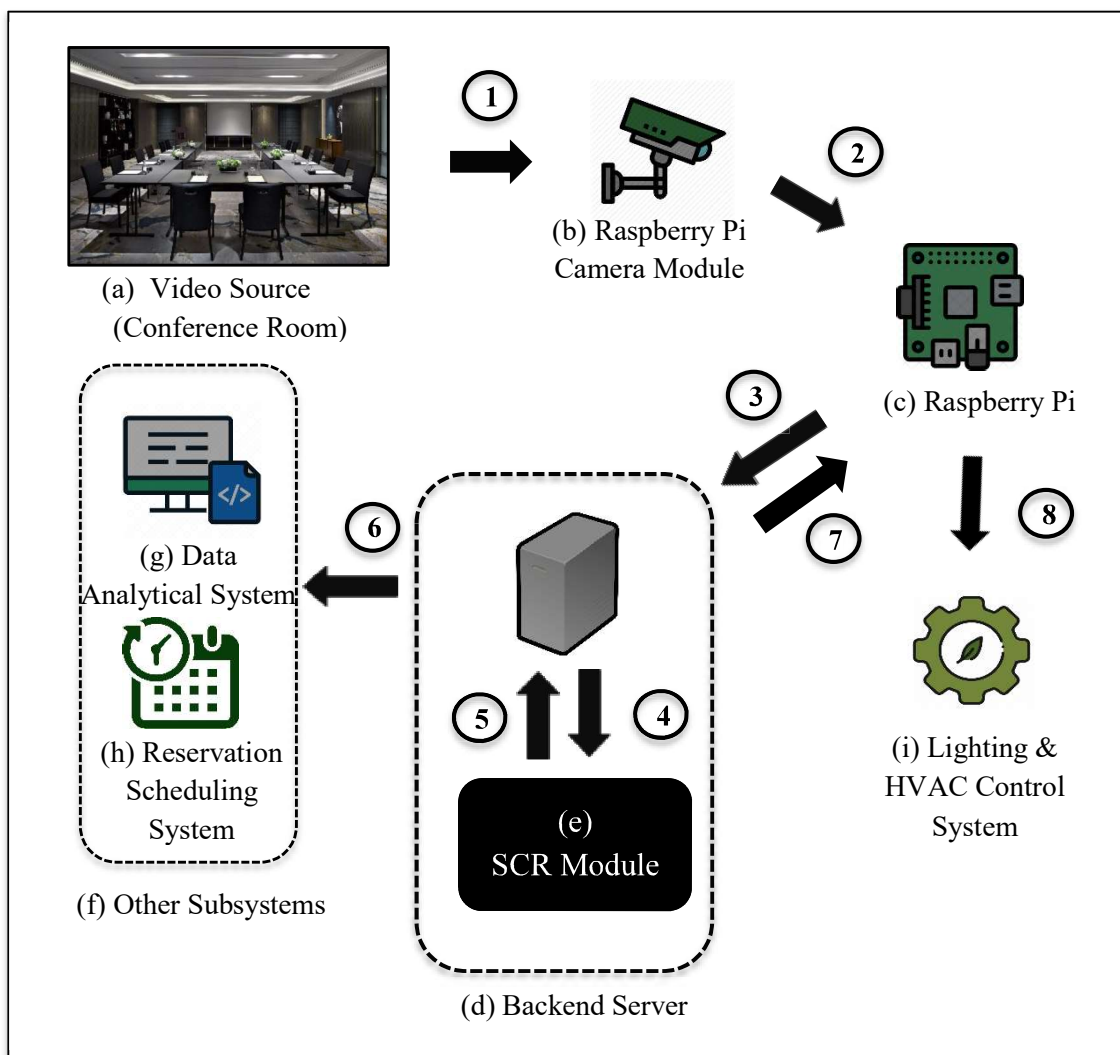
**Figure 2.14 DarkNet-53 architecture (Redmon et al. 2016)**

As YOLOv3 is able to perform prediction in real time manner, it will be used in this project to detect the number of occupants and seats in a conference room for analytical purposes.

## CHAPTER 3 MEETING ROOM EVENT DETECTION

This chapter is organized as follow. Section 3.1 discusses the design and overview of the entire system. Section 3.2 discusses the targeted events to be detected in this system and how the event detection is being performed in this system. Section 3.3 discusses the steps for data preparation before training the model. Section 3.4 discusses the implementation details for training and testing process. Section 3.5 discusses the conducted experiments and their results respectively.

### 3.1 System Overview



**Figure 3.1: System Framework Overview**

The numbering in Figure 3.1 indicated the sequence of system flow.

The flow of the system is shown in Figure 3.1. A Raspberry Pi with attached camera module is installed in the target smart conference room. The Raspberry Pi is loaded with a flask application that is responsible to collect the video into its buffer and send them over the network to the backend server in real time. Then, the video streams received by backend server are loaded into the SCR module.

The SCR module is developed based on deep learning framework that is able to detect and predict the human action. The details about the architecture of SCR module will be discussed later. Then the SCR module will classify and output that the on-going activities in the conference room are either meeting, non-meeting activities or empty. Other than that, the SCR module will perform object detection task in order to track the number of occupants and seats in the conference room. Thenceforth, other subsystems such as data analytical system and reservation scheduling system is able to depend on this information to operate effectively. For instance, the reservation scheduling system will be updated and acknowledged that there is an ongoing meeting event in the particular conference room. The data analytical system is also able to use the information to derive various analytics and generate reports.

Besides, the backend server will respond to Raspberry Pi in order to acknowledge it that there is an ongoing meeting activity. Then, Raspberry Pi will control the lighting and HVAC of the smart conference room accordingly. Else if the conference room is occupied and there are no any meeting activities recognized from SCR module, the status of the conference room will be set as available, then, the reservation scheduling system will be updated. The implementation of the SCR module is discussed below.

## **3.2 Event Detection for SCR**

### **3.2.1 Targeted Events**

In this project, event detection is made possible by the implementation of human action recognition technique. Unlike the occupancy analysis method, event detection method is able to take the context of human action into account to deduce if there's an on-going meeting or in the conference room.

In order to train a reliable model for human action recognition in conferencing environment, this project will have to incorporate of the step to generate the dataset in

conferencing environment as there is still no existing publicly available dataset for meeting activities. Thus, in order to generate this application-specific conference room dataset, it is done in the collaboration with Company X. To ensure the dataset to be unbiased, the footages are collected in the realistic scenario, where there's a surveillance camera in several rooms which constantly record the activities in the conference rooms throughout the day. The footages are recorded from 6 different rooms with different events. The total duration of collected footages after data cleaning are around 200 hours. Data cleaning is performed to remove the inappropriate footages caused by camera misplacement. Then, the dataset is annotated manually by going through the dataset thoroughly. The process will be discussed in section 3.3.

The targeted events to be detected are:

- Meeting  
Meeting event consist of common activities such as standing-up meeting, all-seated meetings, presentation, writing notes, typing on laptop and hand gestures.
- Non-Meeting  
Non-Meeting event consist of activities such as squatting (abuse), private usage (one occupant), room cleaning and room maintenance. Besides, it also consist of activities of occupant entering and leaving the room.
- Empty  
Background class to handle the empty room instances.

Originally, the system is expected to be able to detect more event classes such as cleaning and room maintenance. However, due to the rarity of these events, the dataset only contains scarce amount of these footage which are apparently not sufficient to train a reliable model. Due to this consideration, the requirement of this project has been adapted to compromise with current situation. The classes which have insufficient instances are automatically categorized into their related class. For instance, room cleaning and maintenance are categorized into Non-Meeting event.

Note: Company X is made anonymous at the request of the company itself.

### 3.2.2 Proposed Event Detection Method

In order to detect the event proposed above, SCR module is designed to detect the events for given video inputs. The overall architecture which used for event detection is as shown below.

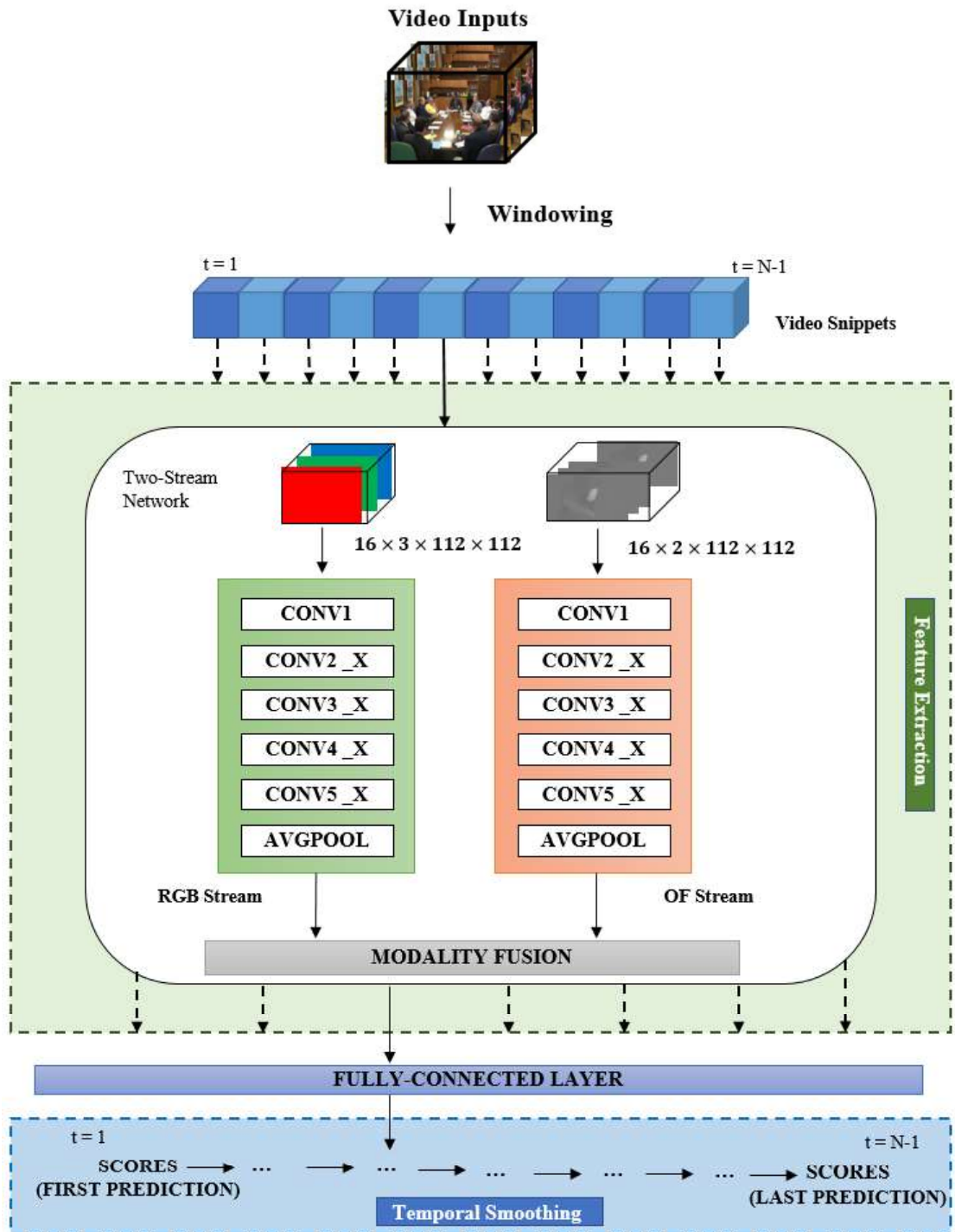
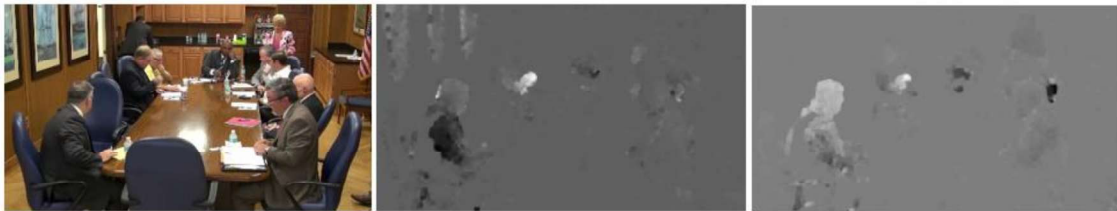


Figure 3.2 Overview of network architecture

**Video Snippets Extraction (Windowing).** Video inputs are applied with windowing technique to chop them into multiple snippets of 16 frames. All of the video snippets will be rescaled to the size of  $320 \times 256$  for dimensionality reduction. Then the video snippets are converted into RGB and optical flow (OF) frames to encode the different modalities. RGB is used to encode the appearance information while OF is used to encode the motion information across the frames. OF frames are computed by using Farneback's algorithm. Farneback's algorithm is chosen due to the reason that it can compute optical flow in real-time while maintaining respectable accuracy. Then, the frames will be further downsized to  $128 \times 171$  and being center cropped to size  $112 \times 112$  before feeding into the network. The output of RGB and OF frames are as shown below.



**Figure 3.3 RGB, Horizontal and Vertical OF frames**

**Feature Extraction.** Both type of the modalities will be handled by a separated network. In order to perform feature extraction, RGB inputs will be fed into RGB stream network while OF inputs will be fed into OF stream network. To illustrate, the inputs of the RGB based R(2+1)D network have size of  $3 \times L \times H \times W$ . 3 indicates the number of channel for RGB inputs. On the other hand, grayscaled OF inputs computed by Farneback's algorithm have only 2 channels. Both of the network will be based on the framework of residual learning as it is able to solve the vanishing gradient. There are different variants of ResNet available. In consideration of the real time implementation and its detection accuracy, the ResNet variants of 18 layers and 152 layers are infeasible for this project. This is due to the reason that the former variant has lower detection accuracy and the latter is more computationally expensive. Consequently, the ResNet with 34 layers setting is the most appropriate option for this project. Then, ResNet-34 will be implemented using (2+1)D convolutional block. That being said, the network architecture of the R(2+1)D with 34 layers variant is shown figures below.



LAYER NAME	OUTPUT SIZE	R(2+1)D-34
conv1	$L \times 56 \times 56$	$3 \times 7 \times 7, 64, \text{stride } 1 \times 2 \times 2$
conv2_x	$L \times 56 \times 56$	$\begin{bmatrix} 3 \times 3 \times 3, 64 \\ 3 \times 3 \times 3, 64 \end{bmatrix} \times 3$
conv3_x	$\frac{L}{2} \times 28 \times 28$	$\begin{bmatrix} 3 \times 3 \times 3, 128 \\ 3 \times 3 \times 3, 128 \end{bmatrix} \times 4$
conv4_x	$\frac{L}{4} \times 14 \times 14$	$\begin{bmatrix} 3 \times 3 \times 3, 256 \\ 3 \times 3 \times 3, 256 \end{bmatrix} \times 6$
conv5_x	$\frac{L}{8} \times 7 \times 7$	$\begin{bmatrix} 3 \times 3 \times 3, 512 \\ 3 \times 3 \times 3, 512 \end{bmatrix} \times 3$
	$1 \times 1 \times 1$	spatiotemporal pooling, FC layer with softmax

Figure 3.4 R(2+1)D-34 architecture

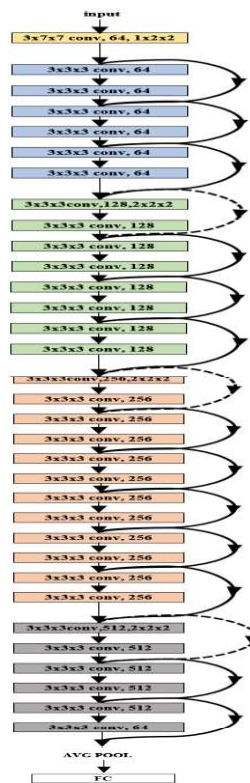


Figure 3.5 R(2+1)D - 34 layers

For each stream, the top layer will performs global average pooling and yields a 512-dimensional feature vector.

**Modality Fusion.** The output of both streams will be 2 feature vectors. Both feature vectors will be fused by averaging and then fed to the final classification (FC) layer. The output dimension of the FC layer will be correspond to the number of classes, which is 3 (Meeting, Non-Meeting and Empty) in this project.

**Temporal Smoothing.** The output of the FC layer will be the softmax scores of each classes. In order to smooth out the predictions and suppress abrupt changes of the predictions, the scores of each class are applied with exponentially weighted moving average (EWMA). EWMA can be formulate as:

$$S_t = ax_t + (1 - a)S_{t-1}$$

$S_t$  and  $x_t$  denotes the moving average and new value at time  $t$ . The power of smoothing can be controlled by setting the value of  $a$ . If  $a$  is set to be small,  $a = 0.1$ , the smoothing will be more powerful as it favors more to previous average,  $S_{t-1}$ . In contrary, it will favors more to the new value,  $x_t$  if  $a$  is larger.

### 3.3 Dataset Generation and Preparation

Prior to train the model proposed above, this section discusses the processes which used to generate and prepare Conference dataset for training.

**Dataset Collection.** Conference dataset is collected in company X. It is collected by setting up a Raspberry Pi with an attached camera module in 6 different conference rooms. Every room varies in layout and lighting conditions. In addition, every room is frequently used by different numbers of occupants from different departments. Every camera is captured from a different angle of view in each room. The raw dataset contains around a total of 200 hours of realistic untrimmed footage with size of (640x480). Each video is around 3 to 6 hours long. The total size of the dataset is around 60GB. Each video can contain multiple events such as Meeting, Non-meeting and Empty. Unlike the others, the Conference dataset requires manual cleaning and annotation. This is the main dataset used for training models to detect conference events.

**Data Cleaning.** Conference dataset is untrimmed and contains various events in one video. Moreover, the dataset contains some flaws which are caused by camera misplacement, blocked sight and corrupted video. Prior to the annotation process, the dataset has been gone through thoroughly to remove unusable videos.

**Data Annotation.** The dataset has been explored completely in order to discover useful labels. Prior to the exploration, the labelling is expected to be finer which may include extra events such as “Room Cleaning”, “Room Maintenance” and “Tele-Conferencing”.

However, coarse labelling is preferred for this dataset as it only contains very few instances of these events. For instance, “Room Cleaning” and “Room Maintenance” are grouped into “Non-Meeting” events and “Tele-Conferencing” is grouped into “Meeting” events. This is due to the reason that there’s only few room cleaning instances discovered after going through the entire dataset. The final labels are chosen to be “Meeting”, “Non-Meeting” and “Empty” as discussed in Chapter 3. The Conference dataset is annotated manually in csv format which includes `video_id`, `start_time`, `end_time` and `label`. Besides, a huge number of instances from the dataset belongs to “Empty” events as the camera is set to record the realistic situation of the conference room without trimming. Hence, the dataset distribution is extremely uneven. In order to compromise for fewer instances from “Meeting” and “Non-Meeting” events, the instances from “Empty” are reduced while still maintaining the balanced distribution from 6 different rooms. The annotation process alone took about 5 weeks in total.

After the annotation, the dataset has been separated into 1128, 577 and 1146 instances of Meeting, Non-Meeting and Empty events. In fact, the dataset contains a scarce amount of “Non-Meeting” instances, which may hurt the performance of the models. In the later experiments, various augmentation techniques will be attempted to reduce the impact of unevenly distributed dataset.

**Data preprocessing.** According to the annotated csv file, a python script is written to extract all the annotated clips from the dataset using FFMPEG. Meanwhile, all the clips are converted into standard 25 frames per second. This is due to the reason that the dataset collected is not in standard FPS, which may lead to abnormal speeding in the clips. In order to feed the dataset into the network, the dataset is being preprocessed and saved to disk prior to training. Each of the videos is trimmed into multiple 6-second chunks (clips) using FFMPEG. Therefore, each clip contains roughly  $6 \times 25$  frames (25FPS). Each video clip is scaled to the size of 320x256. The frames within each clip are then extracted and saved as a sequence of RGB and optical flow (OF) images. OF will be pre-computed using Farneback’s algorithm, thus, resulting in grey-scaled horizontal and vertical optical flow images. Despite the fact that the Farneback algorithm has fast computation time, it will still introduce bottlenecks if it is being computed on-the-fly. Thus, OF frames are precomputed in this stage. As a result, the datasets are preprocessed into 2 separate directories which contain RGB frames and OF

frames. The datasets are stored in frames (JPEG format) according to their frame order. It is then divided into distribution of 60% training, 12% validation and 28% testing.

### 3.4 Experimental Setup

**Training setup.** Due to the great success of (Tran et al. 2018) for pretraining the model on Kinetics, the pretrained weights are loaded for better initialization rather than training from scratch. Based on the Conference dataset as discussed in the previous subsection, the datasets are already preprocessed into frames (in format of JPEG) with size of 320x256 for RGB frames and OF frames. The R(2+1)D architecture will be trained on both stream of inputs (RGB and OF). During the training,  $L=16$  consecutive frames are randomly sampled for temporal jittering. Optimal  $L$  value will be tested again in the experiment later. Batch normalization is added to all the layers. The mini batch size of 4 is used based on the consideration of the hardware resources. Stochastic Gradient Descent (SGD) optimizer is used throughout the experiment. The fine-tuning processes are done in 45 epochs where the model is proven to be able to converge. The total time for the fine-tuning processes on a 4GB GTX 1050Ti are as follows.

Modality	Training Time
RGB	18 hours
OF	18.5 hours

**Table 3.1 Total training time for different modalities**

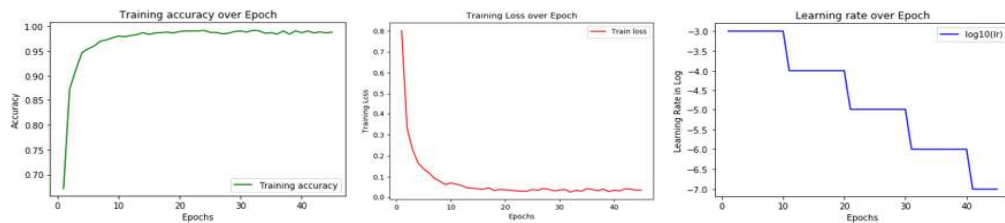
**Testing setup.** Unlike training, center cropping will be used instead of random cropping during testing as data augmentation is not needed in this stage. The model will be tested based on top 1 accuracy only. Conventionally, top 1 and top 5 accuracies are the standard in action recognition domain. However, this dataset consists of 3 classes only, thus, top 5 accuracy is discarded. As a common practice, the model will be tested on clip and video level accuracy. In order to obtain video level accuracy, 10 clips from the video are uniformly sampled by center cropping and the 10 predictions are averaged to get final prediction. In addition, a 10-crop testing method will be used for deeper investigation. The 10-crop testing method is achieved by obtaining a center and 4 corner crops from the clip and the other 5 crops from the horizontal reflections of the clip.

### 3.5 Experiments

This section discusses the experimental results obtained from empirical studies.

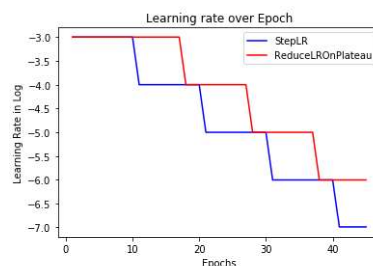
#### 3.5.1 RGB Network

Firstly, transfer learning is applied by loading the weights from pretrained model for better initialization. The network takes  $L = 16$  consecutive frames with size of  $112 \times 112$  as inputs. The learning rate is initialized to 0.01 and is scheduled to be divided by 10 every 10 epoch (step learning rate). The training is done in 45 epochs. The training process is verified by visualizing the training accuracy, training loss and learning rate over epochs. The graphs show that the model is able to converge properly.



**Figure 3.6 Training accuracy, training loss, learning rate over epochs**

Then, further investigation is performed to see if the model will be able to converge better if a different learning rate scheduler is used. Instead of a plain step learning rate scheduler, a smarter scheduling technique (ReduceLROnPlateau) is used. The scheduler will only decrease the learning rate when the optimizer is unable to improve. Generally, if the optimizer reaches a plateau, the learning rate will only be reduced for the optimizer to move down the error surface and approach the minima. The learning rate scheduler is set to wait for 10 epochs (patience). If the model does not improve for 10 epochs, the learning rate will be reduced by a factor of 10. Based on the graph below, it shows that 10 epochs are insufficient to warm up the model to train well. The later scheduler decides it's better to start reducing its learning rate only at the 17-th epoch.



**Figure 3.7 Learning rate over epochs (StepLR/ReduceLROnPlateau)**

Moreover, an experiment is conducted to investigate if the performance will improve if part of the weights from the pretrained models are retained. The experiment is set to retain the weights of earlier layers (before Conv2\_x). The results shown that the model is able to perform better when part of the weights are retained. This may be due to the reason that the pretrained model are capable to provide more useful knowledge in the earlier layers. Besides, another experiment is conducted to determine the best  $L$  selection. The only feasible values of  $L$  for real time detection speed are  $L = 8$  and  $L = 16$  as the Raspberry Pi is set to capture 25 FPS. Hence, the  $L$  selected must be lesser than 25 for millisecond detection speed. Eventually,  $L = 16$  is selected owing to the better accuracy as shown in Table 4.2.

Despite the fact that random cropping and temporal jittering is applied while training, the model is susceptible to be able to generalize well as there is only too few variation in the dataset. The dataset only contains instances from 6 different rooms. In fact, the data point in the dataset might not be sufficient for model to generalize well. Hence, in order to generate more data, the dataset is applied with various augmentation. The applied augmentation is as follow.

- Random horizontal flipping
- Random brightness augmentation
- Random color shifting (on RGB channels)

After the experiments, only random horizontal flipping is tested to be able improve the performance. Besides, in order to reduce the chance for the model to overfit to the training data, the capacity of the model is reduced by applying dropout regularization technique. Dropout regularization is being implemented with a dropout value of  $p = 0.5$  and  $p = 0.9$ . The result turns out to be positive as the accuracies improved. The most significant experimental results obtained are as shown below.

Freezing Point	Clip length	Dropout	Clip Acc.	Video Acc.
None	8	0.0	69.5%	71.3%
None	16	0.0	74.9%	76.6%
Conv2_x	16	0.0	78.6%	82.4%
<b>Conv2_x</b>	<b>16</b>	<b>0.5</b>	<b>83.8%</b>	<b>89.7%</b>
Conv2_x	16	0.9	82.3%	87.4%

**Table 3.2 RGB – Accuracies for experiments**

### 3.5.2 OF Network

Similarly, the model is being trained with the pretrained weights initialization. The learning rate is initialized to be 0.01 and ReduceLROnPlateau is used with patience of 10. After 45 epoch, the model is found that it is unable to converge. After several trials and error, the issue seems to be raised from the sampling rate of the dataset itself. This is due to the reason that the dataset is collected in more than 100 FPS, causing the computed optical flow to be highly inaccurate. The motion differences between frames are too insignificant since the frames are sampled too rapidly. The dataset is re-generated by using the sampling rate of 5. The model is being trained again on the newly generated dataset without any issue.

Unlike the RGB stream, the model perform better when it is able to update its parameter freely without freezing layers. Besides, OF stream is also more sensitive to the clip length as the captured motion information is reliant on the clip length. Generally, the longer the clip length, the higher the density of encoded motion information in the optical flow. The model performs better with  $L = 16$  as shown in the table below.

Moreover, the data augmentation technique used in the training process of RGB stream is tested to be unhelpful in the case of OF stream. This is actually the expected outcome as color and brightness shifting does not affect the optical stream computation. In addition, random horizontal flipping and dropout regularization does not improve the model as well.

Furthermore, it is usually advantageous to zero center the input of the network, as it lets the model to exploit the rectification non-linearities better. Fortunately, the optical flow displacements are naturally zero-centered due to the fact that given a large variety of motion, the probability of the motion moving into one direction is greatly similar to moving into the opposite direction. Thus, the chance of the motion vector to have positive value is equivalent to negative value. However, this is not the case when the global motion is considered. The optical flow between frames can be dominated by a particular movement such as camera motion. In fact, there are various technique can be used to address this issue. For instance, iDT (reviewed in Chapter 2) explicitly compensate the camera motion by estimating the homography and warping the optical flow. However, in this case, the recording devices which are used to collect Conference

dataset are set to be stationary. Therefore, camera motion will not be an issue in this dataset. A simpler approach is used to compensate the effects implicitly, flow mean subtraction is applied by subtracting each motion vector with its mean. The results are shown to be positive by applying flow mean subtraction. The most significant experimental results obtained are as shown below.

<b>Freezing Point</b>	<b>Clip length</b>	<b>Flow Mean Subtraction</b>	<b>Clip Acc.</b>	<b>Video Acc.</b>
None	8	False	73.2%	76.8%
None	16	False	77.8%	81.3%
Conv2_x	16	False	71.3%	72.9%
<b>None</b>	<b>16</b>	<b>True</b>	<b>79.4%</b>	<b>83.7%</b>

**Table 3.3 OF – Accuracies for experiments**

### 3.5.3 Fused Two-Stream Network

The best models from both networks are fused by averaging the scores. Inference is performed to verify if the result of the fusion is positive. Other than clip and video accuracies, the models are tested with 10-crops testing method as well. The final accuracies of RGB, OF and Fused stream are as shown below.

<b>Modality</b>	<b>Clip Acc.</b>	<b>Video Acc.</b>	<b>10-crops Acc.</b>
RGB	83.8%	89.7%	90.8%
OF	79.4%	83.7%	85.4%
Fused	85.2%	90.1%	91.3%

**Table 3.4 Final accuracies of RGB, OF and Fused stream**

In addition, an experiment is carried out to test if the model will perform better if the ‘Non-Meeting’ class is removed. By doing so, this is able to test if the model is capable of detecting the presence of occupants accurately. With the ‘Non-Meeting’ class removed, the model is trying to detect if the input belongs to ‘Occupied’ or ‘Non-Occupied’ only. The final outcome turns out to be the model is able to get the accuracy of 99% on the same dataset. Hence, ‘Non-Meeting’ class is the culprit to worsen the performance of the model. This is due to the reason that the dataset is unevenly distributed. Despite the fact that data augmentation is already applied, the dataset still

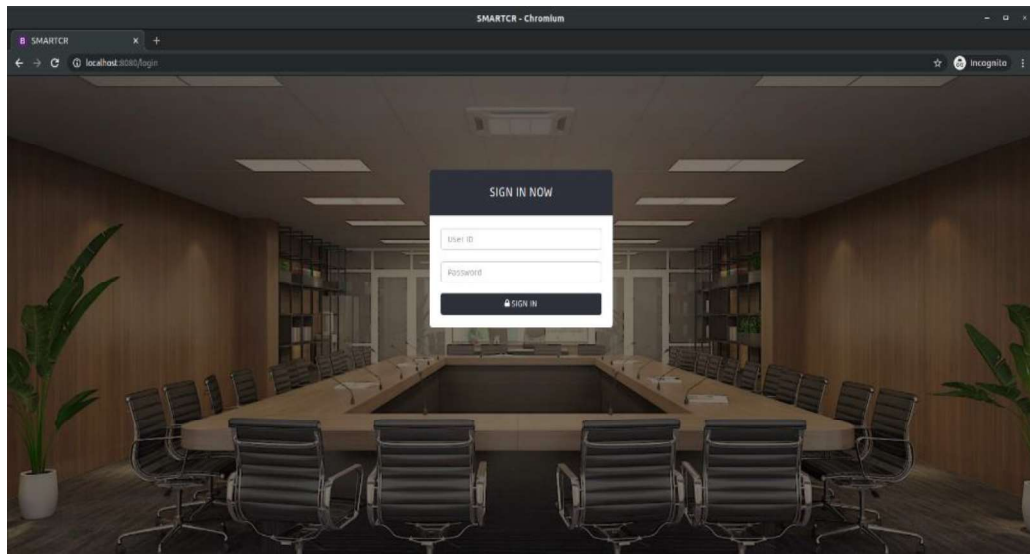


have insufficient instances for Non-Meeting activities. Thus, the model is not able to generalize well.

## CHAPTER 4 WEB APPLICATION DESIGN

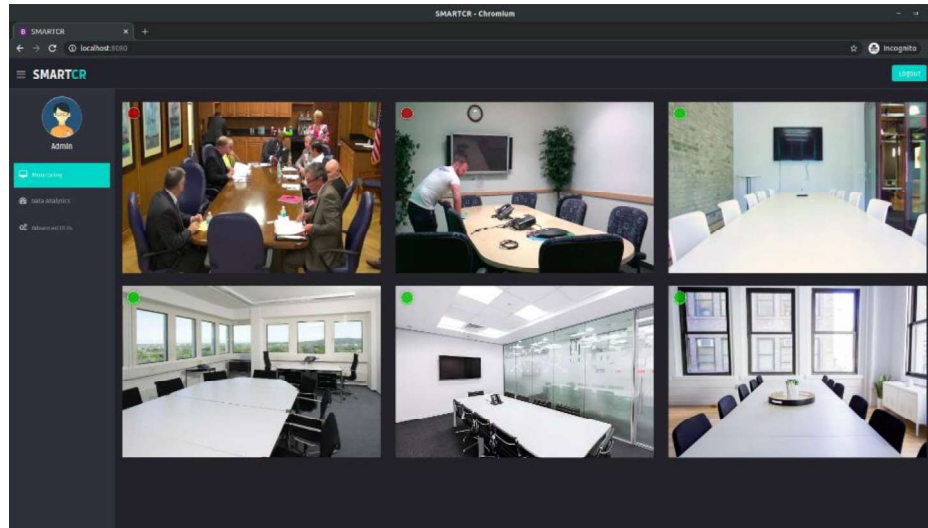
In order to showcase the capabilities of the proposed system, the system is implemented using a simple Flask web application.

**Login page.** The targeted users of this system are the administrators or the managers for the smart conference room. Hence, users will be greeted with a login page, prompting users for administration credentials.



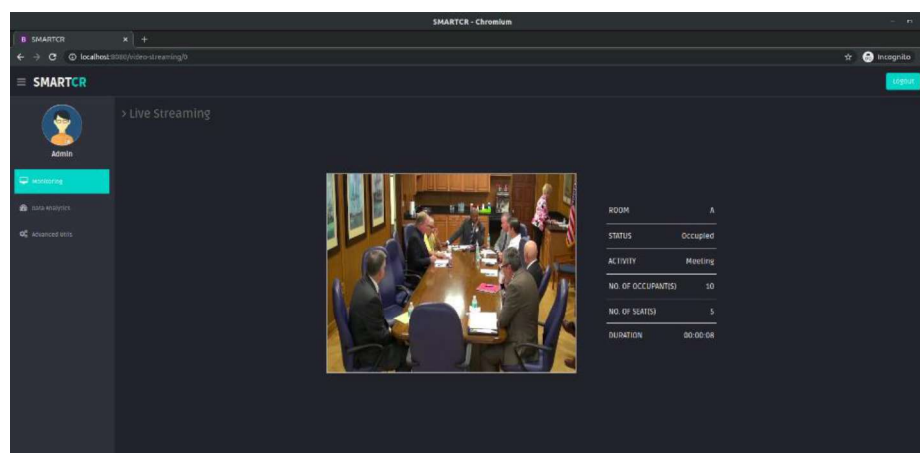
**Figure 4.1 Login Page**

**Administration page.** Then, the system will show users the live streams from all the smart conference room with Raspberry Pi camera installed. However, for demonstration purpose, the system will shows pre-recorded conference videos (obtained from YouTube) instead of recording in live. In fact, the selected pre-recorded conference videos are outside of the dataset themselves, which means that they are not seen by the model in the training process. This is helpful because it can simulate the realistic performance of the system to see if it can generalize well when it is performing detection on the instances which are outside of the dataset. The pre-recorded conference videos representing each room are all paused in the system. This is due to the constraint on computational resource. The system is operating on a machine with GTX 1050Ti GPU, thus, performing detection on multiple rooms simultaneously is impossible. With that being tested, a single 1050Ti GPU can only handle 1 event detection and object detection in parallel before running out of memory. Therefore, the streaming will only starts once the users clicked into the room. The availability of each room will be indicated by the blinking red/green light on the top left corner.

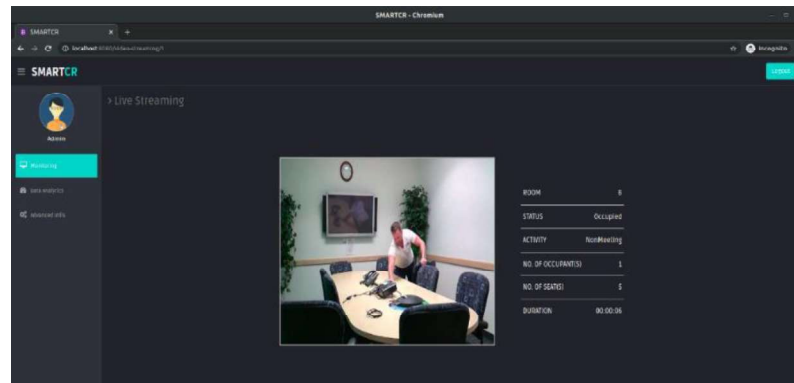


**Figure 4.2 Administration Page (Index)**

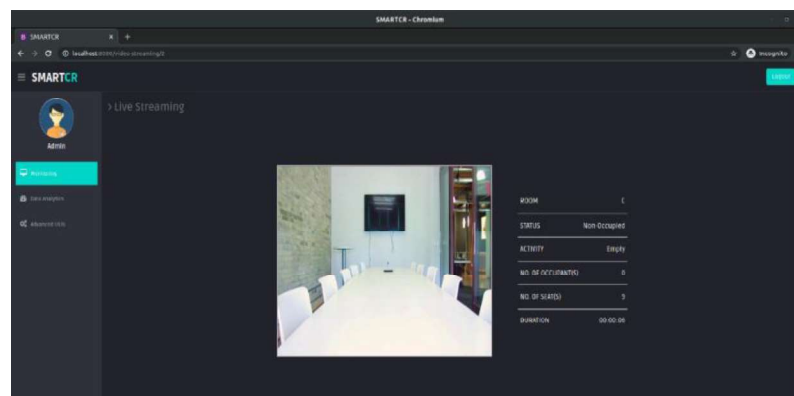
**Monitoring page.** For this demonstration, the event and object detection will only be started once the users clicked into the room. In fact, both of the detection will not be stopped in the real word application. In order to jump start the event detection, 16 frames with sampling rate of 1 will be pulled from the streaming to perform event detection to get the prediction as soon as possible when user clicked into the room. After the first prediction, the sampling rate will be increased to 5 for better sampling. The predicted class confidence scores will be smoothed by applying exponential weighted moving average to avoid abrupt change of the prediction. Object detection will be initiated on the first frame and will continue to predict for each 320 frames (10 seconds). Figure 4.3 (Room A/Meeting) shows the predicted events and number of occupants and seats. The duration will be measured once the Meeting event started. Several example are as shown below.



**Figure 4.3 Monitoring Page for Room A**



**Figure 4.4 Monitoring Page for Room B**



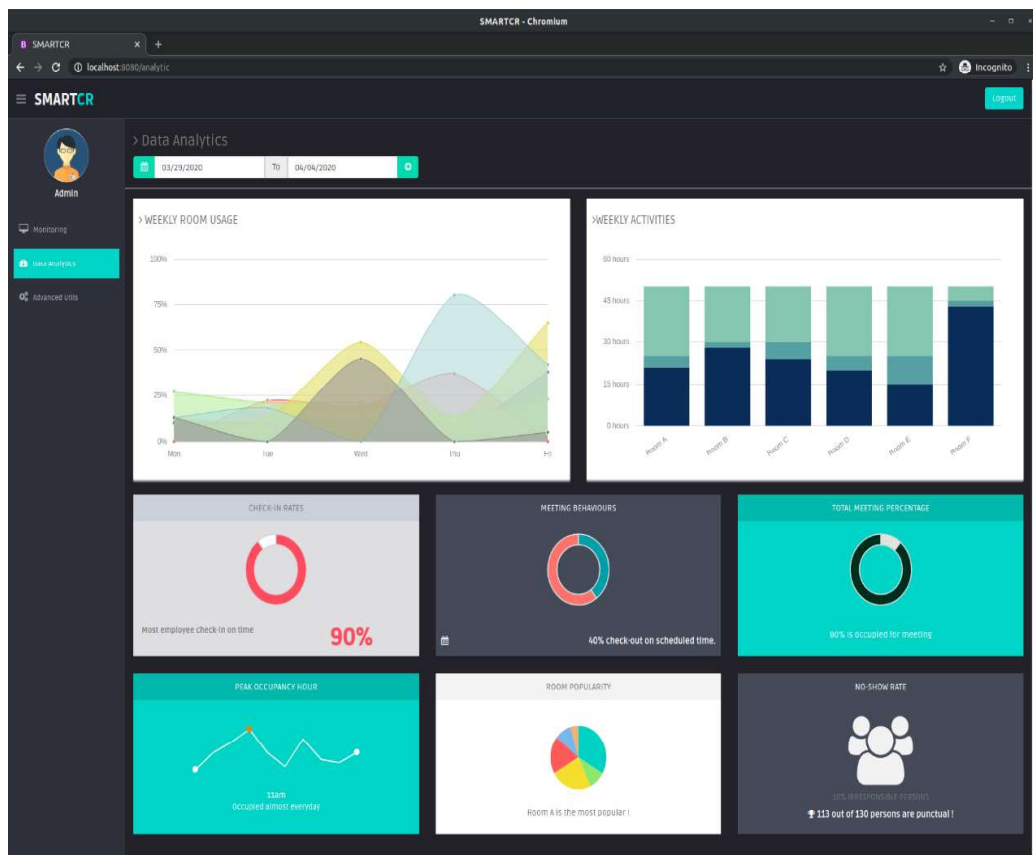
**Figure 4.5 Monitoring Page for Room C**

**Data Analytic Page.** For demonstration purpose, the data analytical page is used to showcase the data analytics or reports that can be derived from the event and object detections. The page can be interacted by changing the time range for the reports. In this demonstration, dummy data is used as placeholder as the machine is incapable of performing detection on all room. However, if the system is implemented in the real world, it should be constantly performing detection, and the detection results can be derived into various analytics. The analytics and reports are shown as such.

- **Weekly Room Usage:** The percentage of the meeting rooms are occupied throughout the week.
- **Weekly Activities:** The ratio of the total time which the rooms are used for ‘Meeting’, ‘Non-Meeting’ and ‘Empty’.
- **Check In Rates:** The percentage of the occupants who check in the meeting room according to predetermined schedule.
- **Meeting Behaviors:** The percentage of the occupants who check out the meeting room according to predetermined schedule.

- **Total Meeting Percentage:** The percentage of all the rooms which are used for ‘Meeting’ purpose only.
- **Peak Occupancy Hour:** The graph for visualizing the frequency of ‘Meeting’ event in hourly manner throughout the selected time range.
- **Room Popularity:** Pie chart shows the room popularity.
- **No-Show Rate:** The percentage of occupants who did not show up for ‘Meeting’ according to pre-determined schedule.

The UI for analytics is as shown in Figure 4.6.



**Figure 4.6 Data Analytics Page**

**Developer Page.** Developer page is used to showcase and visualize how the event detection technique works in the background. It visualizes the RGB, horizontal and vertical optical flow frames before feeding into the network. Besides, it also shows the class confidence score for RGB, OF and Fused stream. The UI is as shown below.

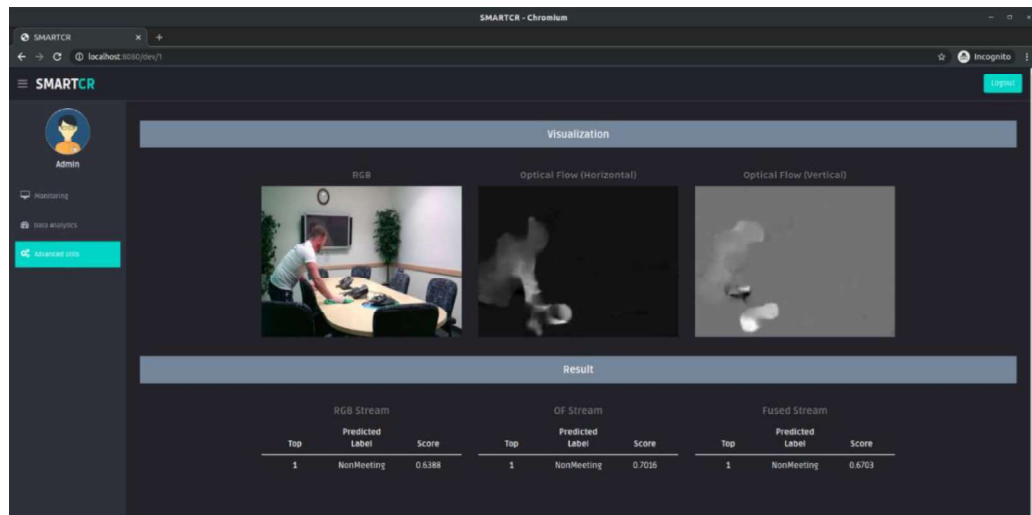


Figure 4.7 Developer Page

## CHAPTER 5 CONCLUSION

In this project, the more recent techniques which based on human action recognition algorithm are used detect the event in a conference room. Unlike the existing works, where majority of them only model the occupancy information via various occupancy analysis techniques, this project proposed a more robust way to handle and detect the different on-going events in a conference room. The motivation of this project is to showcase the feasibility of the implementation of action recognition and object detection techniques in this domain. Meanwhile, different related techniques are inspected in this project. Based on the trade-off between detection speed and accuracy, this project is implemented using two stream network, R(2+1)D action recognition technique and YOLO object detection technique. The idea behind this selection is to allow the system to perform event detection in online and real time settings while not sacrificing too much on the detection accuracy. Meanwhile, object detection is used as an extension to showcase the capabilities of the system by tracking the occupancy rates in the conference room for extra data analytics.

This project also outlined the drawbacks of R(2+1)D in the literature review. As discussed, the design of R(2+1)D is flawed by temporal modelling limitation. In the future, this project may continue to resolve this issue by implementing temporal fusion. This can be done by changing the architecture of R(2+1)D to incorporate recurrent layers such as LSTM to aggregate the long-term features to better model the complete action from the inputs.

We hope that this project is able to help different organizations to facilitate the conferencing activities and workspaces more effectively.

## BIBLIOGRAPHY

- A. Prest, C. Schmid, and V. Ferrari. 'Weakly supervised learning of interactions between humans and objects.' *IEEE PAMI*, 34(3):601–614, 2012.
- Busso, C, Hernanz, S, Chu, C, Kwon, S, Lee, S, Georgiou, P, Cohen, I and Narayanan, S 2005, 'Smart Room: Participant and Speaker Localization and Identification' *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. ii-1117
- Carreira, J & Zisserman, A 2017, 'Quo vadis, action recognition? a new model and the kinetics dataset.', *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6299-6308
- Donahue, J, Anne Hendricks, L, Guadarrama, S, Rohrbach, M, Venugopalan, S, Saenko, K and Darrell, T 2015, 'Long-term recurrent convolutional networks for visual recognition and description.' *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2625-2634
- Feichtenhofer, C., Pinz, A. and Zisserman, A., 2016. 'Convolutional two-stream network fusion for video action recognition.' *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1933-1941
- Ge, Z, Sharma, S, Smith, M 2012, 'PCA/LDA approach for text-independent speaker recognition'. *Independent Component Analyses, Compressive Sampling, Wavelets, Neural Net, Biosystems, and Nanoengineering X*, Vol. 8401, p. 840108.
- Girshick, R 2015, 'Fast r-cnn.' *Proceedings of the IEEE international conference on computer vision*, pp. 1440-1448
- Girshick, R, Donahue, J, Darrell, T & Malik, J 2014, 'Rich feature hierarchies for accurate object detection and semantic segmentation.' *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580-587.
- Greff, K, Srivastava, R, Koutnik, J, Steunebrink, B & Schmidhuber, J 2017, 'LSTM: A Search Space Odyssey.' *IEEE Transactions on Neural Networks and Learning Systems*, 28(10), pp.2222-2232



- Huang, Q 2018, 'Occupancy-Driven Energy-Efficient Buildings Using Audio Processing with Background Sound Cancellation.' *Buildings*, 8(6), p.78
- Huang, Q, Ge, Z & Lu, C 2016, 'Occupancy estimation in smart buildings using audio-processing techniques'
- Ji, S, Xu, W, Yang, M & Yu, K 2013, '3D Convolutional Neural Networks for Human Action Recognition' *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 221-35
- Karpathy, A, Toderici, G, Shetty, S, Leung, T, Sukthankar, R and Fei-Fei, L, 2014. 'Large-scale video classification with convolutional neural networks.' *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition* pp. 1725-1732
- Kay, W, Carreira, J, Simonyan, K, Zhang, B, Hillier, C, Vijayanarasimhan, S, Viola, F, Green, T, Back, T, Natsev, A, Suleyman, M, & Zisserman, A 2017, 'The Kinetics Human Action Video Dataset'
- Klaser, A, Marszałek, M & Schmid, C 2008, 'A spatio-temporal descriptor based on 3d-gradients.'
- Laptev, I, Marszalek, M, Schmid, C & Rozenfeld, B 2008, 'Learning realistic human actions from movies.' *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-8
- Redmon, J & Farhadi, A 2018, 'YOLOv3: An Incremental Improvement'.
- Ren, S, He, K, Girshick, R & Sun, J 2015. 'Faster r-cnn. Towards real-time object detection with region proposal networks.' *In Advances in neural information processing systems*, pp. 91-99
- Scovanner, P, Ali, S & Shah, M 2007, 'A 3-dimensional sift descriptor and its application to action recognition.' *Proceedings of the 15th ACM international conference on Multimedia*, pp. 357-360
- Simonyan, K & Zisserman, A 2014, 'Two-stream Convolutional Networks for Action Recognition in Videos', *Proceedings of the 27th International Conference on Neural Information Processing Systems*, pp. 568-576

- Soomro, K, Zamir, A & Shah, M 2012, 'UCF101: A Dataset of 101 Human Actions Classes from Videos in The Wild'
- Tran, D, Bourdev, L, Fergus, R, Torresani, L & Paluri, M 2015, 'Learning Spatiotemporal Features with 3D Convolutional Networks.' *Proceedings of the IEEE international conference on computer vision*, pp. 4489-4497
- Tran, D, Wang, H, Torresani, L, Ray, J, LeCun, Y & Paluri, M 2018, 'A Closer Look at Spatiotemporal Convolutions for Action Recognition'. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6450-6459
- Wang, H & Schmid, C 2013. 'Action recognition with improved trajectories.' *Proceedings of the IEEE international conference on computer vision*, pp. 3551-3558
- Wang, H, Kläser, A, Schmid, C & Liu, CL 2011 'Action recognition by dense trajectories.' *CVPR 2011*, pp. 3169-3176
- Zou, H, Zhou, Y, Jiang, H, Chien, S, Xie, L and Spanos, C 2018, 'WinLight: A WiFi-based occupancy-driven lighting control system for smart building'. *Energy and Buildings*, pp.924-938

# Event Detection for Smart Conference Room using Spatio-Temporal Convolutional Neural Network

Tan Yi Jian, Universiti Tunku Abdul Rahman

## POSTER

### Introduction

There are various attempts on building smart conference room in the past. This is the first work that attempt to detect the on-going event in a conference room based on human action recognition.

### Problems:

Without event detection, the system is unable to provide reliable monitoring information of the conference room.

### Solutions:

- Build a real-time system which is based on human action recognition technique that is able to accurately detect event based on human action.
- Prepare and preprocess datasets in conferencing environment.
- Develop data analytics tools using object detection technique

### System Design:

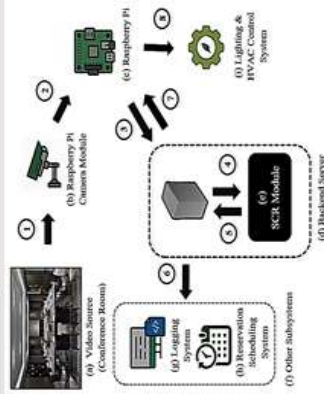


Figure 1: Overview of system design.

### Network Architecture Overview:

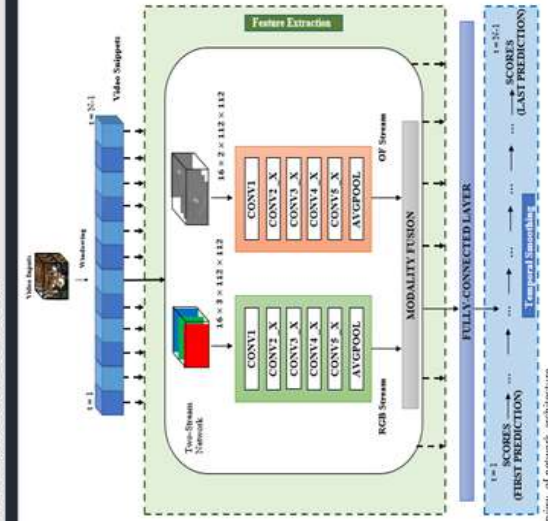


Figure 2: Overview of network architecture

### R(2+1)D – 34 Layers Network Architecture

LAYER NAME	OBJECT SIZE	R(2+1)D
conv1	4 × 56 × 56	3 × 7 × 7, stride 1 × 2 × 2
conv1_3	4 × 56 × 56	3 × 3 × 3, stride 1 × 2 × 2
conv1_5	2 × 28 × 28	3 × 3 × 3, stride 1 × 2 × 2
conv1_7	2 × 14 × 14	3 × 3 × 3, stride 1 × 2 × 2
conv1_9	2 × 7 × 7	3 × 3 × 3, stride 1 × 2 × 2
conv1_11	1 × 1 × 1	spatiotemporal pooling, FC layer with softmax

Figure 3: R(2+1)D – 34 layers network architecture [Iran et al., 2018].

### Event Detection

The model should be able to detect event:

- Meeting
- Non-Meeting
- Empty

### Experiments

- Dataset:**
  - Kinetics-400 (Pretraining)
  - Contains 400 action classes with at least 400 clips per action class.
  - Conference Dataset (Finetuning)
  - Self generated dataset with 3 classes (Meeting, Non-Meeting, Empty).
- Result:**
  - Evaluated on Conference Dataset:
  - RGB Stream

Freezing Point	Clip length	Dropout	Clip Acc.	Video Acc.
None	8	0.0	69.5%	71.3%
None	16	0.0	74.9%	76.6%
Conv2_x	16	0.0	78.6%	82.4%
Conv2_x	16	0.5	83.8%	89.7%
Conv2_x	16	0.9	82.3%	87.4%

Figure 4: RGB stream accuracies

### OF Stream

Freezing Point	Clip length	Flow Mean Subtraction	Clip Acc.	Video Acc.
None	8	False	73.2%	76.8%
None	16	False	77.8%	81.3%
Conv2_x	16	False	71.3%	72.9%
None	16	True	79.4%	83.7%

Figure 5: OF stream accuracies

### Fused Stream

Modality	Clip Acc.	Video Acc.	10-crops Acc.
RGB	83.8%	89.7%	90.8%
OF	79.4%	83.7%	85.4%
Fused	85.2%	90.1%	91.3%

Figure 6: RGB, OF and Fused stream accuracies

### Conclusion

In this implementation, the system proposed is able to perform real time event detection yet resulting in reliable accuracy.

# PLAGIARISM CHECK RESULT



## Turnitin Originality Report

FYP2V3 by Yi Jian Tan

From FYP submissions (FYP Projects)

Processed on 24-Apr-2020 06:20 +08

ID: 1305656842

Word Count: 12052

Similarity Index		Similarity by Source	
<b>2%</b>		Internet Sources:	0%
		Publications:	1%
		Student Papers:	1%

### sources:

- 1 1% match (publications)  
[Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, Manohar Paluri. "A Closer Look at Spatiotemporal Convolutions for Action Recognition", 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018](#)

---

- 2 < 1% match (publications)  
[Joao Carreira, Andrew Zisserman. "Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset", 2017 IEEE Conference on Computer Vision and Pattern Recognition \(CVPR\), 2017](#)

---

- 3 < 1% match (student papers from 16-Aug-2019)  
[Submitted to Radboud Universiteit Nijmegen on 2019-08-16](#)

---

- 4 < 1% match (Internet from 10-Oct-2019)  
<https://link.springer.com/article/10.1007%2Fs11548-019-01995-1>

---

- 5 < 1% match (Internet from 10-Jul-2017)  
<https://linknovate.com/affiliation/lehigh-university-2194/all/?query=statistical+recognition>

---

- 6 < 1% match (student papers from 22-Nov-2019)  
[Submitted to University Tun Hussein Onn Malaysia on 2019-11-22](#)

---

- 7 < 1% match (student papers from 22-Apr-2006)  
[Submitted to The Hong Kong Polytechnic University on 2006-04-22](#)

---

- 8 < 1% match (Internet from 07-Sep-2017)  
[https://kuscholarworks.ku.edu/bitstream/handle/1808/10466/Fei\\_ku\\_0099D\\_12438\\_DATA\\_1.pdf?isAllowed=y&sequence=1](https://kuscholarworks.ku.edu/bitstream/handle/1808/10466/Fei_ku_0099D_12438_DATA_1.pdf?isAllowed=y&sequence=1)

---

- 9 < 1% match (student papers from 13-Sep-2019)  
[Submitted to Imperial College of Science, Technology and Medicine on 2019-09-13](#)

<b>Universiti Tunku Abdul Rahman</b>			
<b>Form Title : Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)</b>			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 01/10/2013	Page No.: 1 of 1



**FACULTY OF INFORMATION AND COMMUNICATION  
TECHNOLOGY**

<b>Full Name(s) of Candidate(s)</b>	TAN YI JIAN
<b>ID Number(s)</b>	16ACB01606
<b>Programme / Course</b>	BACHELOR OF COMPUTER SCIENCE (HONS)
<b>Title of Final Year Project</b>	EVENT DETECTION FOR SMART CONFERENCE ROOM USING SPATIO-TEMPORAL CONVOLUTIONAL NEURAL NETWORK

<b>Similarity</b>	<b>Supervisor's Comments (Compulsory if parameters of originality exceeds the limits approved by UTAR)</b>
<b>Overall similarity index:</b> <u>  2  </u> %  <b>Similarity by source</b> Internet Sources: <u>  0  </u> % Publications: <u>  1  </u> % Student Papers: <u>  1  </u> %	
Number of individual sources listed of more than 3% similarity: <u>  0  </u>	
<b>Parameters of originality required and limits approved by UTAR are as Follows:</b> (i) Overall similarity index is 20% and below, and (ii) Matching of individual sources listed must be less than 3% each, and (iii) Matching texts in continuous block must not exceed 8 words <i>Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.</i>	

Note Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

***Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.***

\_\_\_\_\_  
Signature of Supervisor  
Name:   Tan Hung Khoon    
Date:   24 April 2020  

-

\_\_\_\_\_  
Signature of Co-Supervisor  
Name:   -    
Date:   -




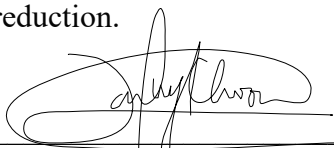
**UNIVERSITI TUNKU ABDUL RAHMAN**  
**FACULTY OF INFORMATION & COMMUNICATION**  
**TECHNOLOGY (KAMPAR CAMPUS)**

**CHECKLIST FOR FYP2 THESIS SUBMISSION**

Student Id	16ACB01606
Student Name	TAN YI JIAN
Supervisor Name	DR TAN HUNG KHOON

TICK (✓)	DOCUMENT ITEMS
	Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
✓	Front Cover
✓	Signed Report Status Declaration Form
✓	Title Page
✓	Signed form of the Declaration of Originality
✓	Acknowledgement
✓	Abstract
✓	Table of Contents
✓	List of Figures (if applicable)
✓	List of Tables (if applicable)
✓	List of Symbols (if applicable)
✓	List of Abbreviations (if applicable)
✓	Chapters / Content
✓	Bibliography (or References)
✓	All references in bibliography are cited in the thesis, especially in the chapter of literature review
✓	Appendices (if applicable)
✓	Poster
✓	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)

\*Include this form (checklist) in the thesis (Bind together as the last page)

<p>I, the author, have checked and confirmed all the items listed in the table are included in my report.</p> <p style="text-align: center;"></p> <p>_____          (Signature of Student)          Date: 22/04/2020</p>	<p>Supervisor verification. Report with incorrect format can get 5 mark (1 grade) reduction.</p> <p style="text-align: center;"></p> <p>_____          (Signature of Supervisor)          Date: 24 April 2020</p>
---	---