

**PERVASIVE IOT: AUTO CALIBRATION IN LIGHT SENSOR USING  
EMBEDDED ACTIVITY RECOGNITION**

**BY  
NEOW ZHI CHIN**

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

**JAN 2021**

UNIVERSITI TUNKU ABDUL RAHMAN

## REPORT STATUS DECLARATION FORM

**Title:** PERVASIVE IOT: AUTO CALIBRATION IN LIGHT SENSOR USING  
EMBEDDED ACTIVITY RECOGNITION

**Academic Session:** JAN 2021

I NEOW ZHI CHIN  
(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.



(Author's signature)

Verified by,



(Supervisor's signature)

**Address:**

B-7-3A, Jalan Bukit Minyak Permai 2,  
Jalan Bukit Minyak, 14000 Bukit Minyak,  
Pulau Pinang

Dr. Aun YiChiet  
Supervisor's name

**Date:** 15 April 2021

**Date:** 15 April 2021

**PERVASIVE IOT: AUTO CALIBRATION IN LIGHT SENSOR USING  
EMBEDDED ACTIVITY RECOGNITION**

**BY**

**NEOW ZHI CHIN**

**A REPORT**

**SUBMITTED TO**

**Universiti Tunku Abdul Rahman**

**in partial fulfillment of the requirements**

**for the degree of**

**BACHELOR OF COMPUTER SCIENCE (HONOURS)**

**Faculty of Information and Communication Technology  
(Kampar Campus)**

**JAN 2021**

## DECLARATION OF ORIGINALITY

I declare that this report entitled “**PERVASIVE IOT: AUTO CALIBRATION INLIGHT SENSOR USING EMBEDDED ACTIVITY RECOGNITION**” 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 : \_\_\_\_\_Neow Zhi Chin\_\_\_\_\_

Date : \_\_\_\_\_15 April 2021\_\_\_\_\_

## **ACKNOWLEDGEMENTS**

First of all, I would like to express my deep gratitude and appreciation to my supervisor, Dr. Aun Yichiet who has given me this bright opportunity to engage in an AI design project. It is my first step to establish a career in AI design field. I will work hard for this project. A million thanks to you.

## **ABSTRACT**

In the era of technology, lighting has become responsive and intelligent. When lighting is combined with internet of things (IoT), it could uphold greater innovated functionality. The emerging of smart lighting that support automatically switch on and off with additional embedded sensors and remotely control of light intensity has bring a comfortable and convenient lifestyle. However, the effect of color temperature on humans are neglected when the advancement of smart lighting is focusing on bringing convenient to humans. An incorrect application of colour temperature will lead to loss of productivity, damage vision ability and affect the well-beings. To address the issue, an intelligent smart lighting system that could change the lighting condition to the optimum colour temperature based on variety of activities is essential. The contribution of this project is to maximize the productivity of human in doing daily task, improved health, and protect the eye vision ability while replacing the conventional way of tuning the color temperature to autonomous.

In this project, a smart lighting system that able to change the light to optimal color temperature using embedded activity recognition is proposed. The dataset that used in this project is Moment in Time Dataset. A CNN-LSTM model that combining VGG-16 as a feature extractor and LSTM as classifier is proposed to perform activity recognition including dining, playing guitar as well as studying. The experimental results show that the proposed model behave well and achieves 75.45 % of precision and 77.76% of recall.

# TABLE OF CONTENTS

<b>TITLE PAGE</b>	<b>i</b>
<b>ACKNOWLEDGEMENTS</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 SYMBOLS</b>	<b>ix</b>
<b>LIST OF ABBREVIATIONS</b>	<b>x</b>
<b>Chapter 1: Introduction</b>	<b>1</b>
<b>1.1 Problem Statement and Motivation</b>	<b>1</b>
<b>1.2 Project Scope</b>	<b>1</b>
<b>1.3 Project Objective</b>	<b>2</b>
<b>1.4 Highlights</b>	<b>2</b>
<b>1.5 Background Information</b>	<b>3</b>
<b>1.5.1 Convolutional Neural Network (CNN)</b>	<b>3</b>
<b>1.5.2 Recurrent Neural Network (RNN)</b>	<b>4</b>
<b>1.5.3 Color temperature</b>	<b>5</b>
<b>1.5.4 Data Augmentation</b>	<b>6</b>
<b>1.6 Report Organization</b>	<b>6</b>
<b>Chapter 2 Literature Review</b>	<b>7</b>
<b>2.1 CNN Transfer Learning</b>	<b>7</b>
<b>2.1.1 Comparison between different pre-trained model</b>	<b>7</b>
<b>2.2 Action Recognition in Video Sequences using Deep Bi-Directional LSTM With CNN Features</b>	<b>9</b>
<b>2.3 Action Classification Based on 3D Convolutional Networks</b>	<b>10</b>
<b>2.4.1 Intelligent human-centric lighting for mental wellbeing improvement</b>	<b>11</b>
<b>Chapter 3: Proposed Method/Approach</b>	<b>13</b>
<b>3.1 Methodology</b>	<b>13</b>
<b>3.2 Tools and technologies used</b>	<b>15</b>
<b>3.3 Standard evaluation metrics for model performance</b>	<b>15</b>
<b>3.4 Dataset Collection</b>	<b>16</b>
<b>3.5 Data Pre-processing</b>	<b>18</b>
<b>3.6 Proposed Network Architecture</b>	<b>19</b>

3.6.1 Fine-Tuned VGG-16 Architecture	19
3.6.2 LSTM Model Architecture	21
3.7 IoT Framework for Activity Recognition	22
Chapter 4: Experimental and Evaluation	23
4.1 Experiment Setup	23
4.2 Jetson Nano Configuration	24
4.3 Smart LED Bulb Configuration	24
4.4 Baseline Model Evaluation	25
4.5 Hyperparameter Optimization	26
4.6 Evaluation on Testing set	28
4.7 Implementation of Jetson Nano	30
4.8 Final Product Evaluation	31
Chapter 5: Conclusion	33
5.1 Project Review	33
5.2 Future Work	33
BIBLIOGRAPHY	34
APPENDIX A: Poster	A-1
APPENDIX B: Final Year Project Weekly Report	B-1
APPENDIX C: Plagiarism Check Result	C-1
APPENDIX D – TURNITIN FORM	D-1
APPENDIX E – CHECKLIST	E-1



# LIST OF FIGURES

<b>Figure Number</b>	<b>Title</b>	<b>Page</b>
Figure 1.5.1.1	Convolutional Neural Network Architecture	3
Figure 2.1.1.1	Inception Layers	8
Figure 2.2.1	External Structure of proposed DB-LSTM network	9
Figure 2.2.2	Internal Structure of proposed DB-LSTM network	10
Figure 2.4.1.1	System scheme	12
Figure 2.4.1.2	Flow of System	12
Figure 3.1.1	Research Methodology	13
Figure 3.4.1	Dataset Directory Structure	17
Figure 3.4.2	Training Annotation File	17
Figure 3.5.1	Frames Extraction Process	18
Figure 3.6.1	Proposed CNN-LSTM Network Architecture	19
Figure 3.6.1.1	VGG-16 Architecture	21
Figure 3.6.2.1	LSTM architecture	21
Figure 3.7.1	Flow Chart for Auto Tuning Light Colour Temperature	22
Figure 4.4.1	Accuracy-Loss Chart	25
Figure 4.5.1	Accuracy-Loss Chart for Best Model	28
Figure 4.6.1	Confusion Matrix	29
Figure 4.8.1	Data regarding of productivity	31
Figure 4.8.2	Data regarding of health	32
Figure 4.8.3	Data regarding of living quality	32

## LIST OF TABLES

<b>Table Number</b>	<b>Title</b>	<b>Page</b>
Table 4.1.1	Library Version	23
Table 4.1.2	LSTM Model Default Parameter	24
Table 4.5.1	Result for different model parameters	26
Table 4.6.1	Overall Performance of The Model	29
Table 4.6.2	Performance metric for each class	30
Table 4.7.1	Optimal Colour Temperature for each activity	30

## LIST OF SYMBOLS

$H(x)$	Desired Function
$F(x)$	Learning Problem

## LIST OF ABBREVIATIONS

<i>CCT</i>	Correlated Colour Temperature
<i>LED</i>	Light Emitting Diode
<i>DNN</i>	Deep Neural Network
<i>DBN</i>	Deep Belief Network
<i>HAR</i>	Human Activity Recognition
<i>ML</i>	Machine Learning
<i>AI</i>	Artificial Intelligence
<i>IoT</i>	Internet of Things
<i>ILSVRC</i>	ImageNet Large Scale Visual Recognition Challenge
<i>RF</i>	Radio Frequency
<i>RBM</i>	Restricted Boltzmann Machines
<i>CNN</i>	Convolutional Neural Network
<i>RNN</i>	Recurrent Neural Network

### **Chapter 1: Introduction**

#### **1.1 Problem Statement and Motivation**

The Internet of Things (IoT) is a hot issue nowadays and the rapid development of the IoT has promoted smart lighting coming into our lives. A lot of ways have been introduced in controlling the lights such as remotely adjust the light brightness. Nevertheless, until today, smart lighting system that capable of adjusting colour temperature automatically based on human activities without requires manual intervention is still unknown. Since there is no existing method that can change the colour temperature automatically based on the current activity, this gave birth to the new idea of developing a smart lighting system using embedded activity recognition. The motivation of this work is that the optimal colour temperature is significant and beneficial to health, productivity and visual acuity of humans. The correct application of colour temperature in an indoor environment can increases productivity, supports cognitive processes and enhance health.

Besides, in the area of activity recognition, most of the training and inference in these days happens in the cloud. There exists severe disadvantage to this approach which is the delay of inference processing during a network congestion or when the cloud servers are overloaded. To cope with this issue, performing deep inference on local device is a better solution. In the local components, the recognition engine is located on the device, so uploading data up to the cloud for processing is not required. Prediction occurs instantaneously on the device without the need to send request over internet and wait for reply. Thus, doing inference on device can be a lot faster and more reliable than doing network requests.

#### **1.2 Project Scope**

This project aims to build a smart lighting system that provide automatic calibration of light colour temperature based on human activity. This system should be able to recognize the activities performed by humans in indoor environment by using embedded activity recognition and change the colour temperature to the optimal lighting condition without manual intervention to enhance human performance and productivity.

### 1.3 Project Objective

The objectives of this project are:

- i. To build an autonomous smart lighting system that can change colour temperature based on human activity with embedded activity recognition.
- ii. To eliminate delayed inference by running inference computations locally on the device without sending request and waiting cloud-based service to response.
- iii. To evaluate the accuracy of activity recognition.
- iv. To enhance the human productivity in performing daily task by creating optimum lighting condition.
- v. To provide automatic color temperature tunable lighting designed to improve human health and wellness.
- vi. To provide a level of comfort and convenience as well as increase quality of living.

### 1.4 Highlights

The outcome of this project is a novel light system that is capable of recognize different indoor activities and auto calibrate the light color temperature to the optimum level that best suits the activity that is recognized. This light system strives to boost the human productivity in performing daily task and enhance health by providing the suitable color condition. Previous studies have found that working under blue enriched light condition can improve work performance by supporting mental acuity and alertness while reducing sleepiness. On the other hand, a low color temperature triggers the release of Melatonin in human body that helps to relax our mind after working for a day and prepare for sleep. Furthermore, this light system is convenient to human as it does not require manual intervention in changing color temperature.

Another contribution of this project is solving the delayed inference by changing inference computations on cloud to run locally on device. This will eliminate the need for uploading the data to cloud to process and wait for reply especially when the network is congested. Hence, the time for recognizing the activity is greatly reduced.

## 1.5 Background Information

### 1.5.1 Convolutional Neural Network (CNN)

In deep learning, Convolution Neural Network (CNN) is a well-known class of deep neural network that composed of neurons that have learnable weights and biases. CNN is usually used to perform image classification and activity recognition because of its high accuracy. CNN functions by extracting features from images which means that the need for manual feature extraction is not required. These extracted features are learned through the hidden layers instead of being trained. The complexity of the learned features increases with the increment of layers. CNN consists of three main layer which are convolutional layer, pooling layer and fully connected layer. The convolutional layer is the major building blocks where filters are applied to the original input images and produce feature maps that can used to detect different features. Convolution will preserve or retain the spatial relationship between pixels even though the size of image is reduced after applying filter. Pooling layer is also known as subsampling where the size or dimensionality of feature map is reduced while retaining the most important features. There are three type of pooling that can apply which are Max, Average and Sum. Fully connected layers on the other hand connect all nodes in one layer to every neuron in the next layer. Figure below depicts the underlying architecture of CNN.

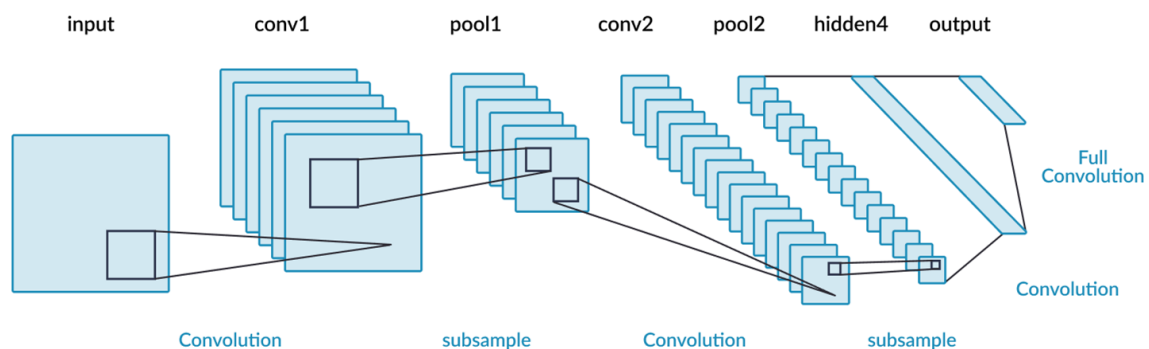


Figure1.5.1.1: Convolutional Neural Network Architecture

### 1.5.2 Recurrent Neural Network (RNN)

Recurrent neural network (RNN) is an artificial neural network that works with time series data. RNN is useful for temporal problems, such as speech recognition, video recognition and image captioning. While conventional deep neural networks presume that inputs and outputs are independent of one another, output of RNN depends on the prior elements of the sequence. RNN is made up of hidden states of high dimensional with non-linear dynamics. The structure of hidden states serves as network's memory and state of the hidden layer at any moment is based on its previous state and hence enables the RNNs to store, remember, and process past complex signals for long period of time. RNNs able to map an input sequence to the output sequence at the current timestep and predict the sequence in the next timestep.

Recurrent networks are often distinguished by the fact that their parameters are shared across all layers in the network. As compare to feedforward network that having a different weight across each node, RNN share the same weight parameter within each layer. However, these weights are still adjusted via the processes of backpropagation and gradient descent to allow reinforcement learning.

To determine the gradient, backpropagation through time (BPTT) algorithm is applied by RNN. It is different from the conventional backpropagation as RNN is specific to data sequence. The concepts of BPTT are similar to conventional backpropagation, in which the model trains itself by measuring errors from the output layer to the input layer. Since feedforward networks do not exchange parameters across layers, BPTT is different from the conventional way that it sums the errors for each time step, while sums errors is not required in feedforward network. During this process, RNN has two common issues which are vanishing gradient and exploding gradient. These problems arise from slope of the loss function along the error curve. If the gradient is very small, it will continue become smaller, updating the weight parameters until they are significant which is zero. When this happens, algorithm will stop learning. On the other hand, exploding gradients happens when the gradient is too large. In this scenario, the model weight will continue become larger and eventually become NaN. A simple solution to these problems is to reduce the number of hidden layers of the network, thus eliminating the complexity in RNN.



### 1.5.3 Color temperature

Colour temperature is a description of the coolness or warmth of a light source and it is used to characterize the type of white light emitted. To define in technical term, it is the temperature at which a theoretical black-body radiator emits light of the same colour to the light source. Colour temperature is measured in Kelvin (K) with higher numbers corresponding to cooler tones and lower numbers corresponding to warmer tones. This is because an increase in the colour temperature will decrease the red components of the light while the blue components increase with increment of light colour temperature. The colour temperature of a light source is assigned using the basis of correlated colour temperature (CCT) and it is usually range from 2700 K to 6500 K. By lighting convention, lights that over 5000K appear to be bluish-white that mimics daylight and is considered cool while light which is under 5000K has a yellow-red colour and is considered warm.

Naturally, the color temperature of lights will have an important impact on how and where you decide to use them. The correct application of CCT in an indoor environment can increase human's productivity, supports cognitive processes and enhance health. Brighter white light in the range of 3500K to 4100K should be kept in workspaces. Studies have shown that cooler light temperatures can improve alertness and productivity. The reason is blue light has the effect of lowering the secretion of melatonin in our glands to keep people to stay alert and reducing fatigue. On the other hand, warmer tones tend to create a sense of relax and comfort, and this kind of lighting is usually used in bedroom to promote sleepiness.

The improper application of CCT might has vital negative consequences on human circadian system and transform mood (Mills, Tomkins and Schlangen, 2007) and these can further lead to loss of productivity (Shamsul et. al., 2013). In order to ensure the daily task can be performed effectively and productively, suitable colour temperature is necessary.

### **1.5.4 Data Augmentation**

Data augmentation is a one of the ways that can prevent overfitting by increasing the sample size of data collected for training models. Data augmentation is usually applied when the dataset is not sufficient. It diversifies the data already available with techniques from Computer Vision and Image Processing. In Layman's term, it is a way to get more data from the data we have right now. To achieve this, the dataset of images could be altered in the way to get different orientation, location, scale and brightness. This will also help the convolutional neural network (CNN) to learn to identify the objects better even if the objects are shown differently or is in different orientation, affected by noises or so on. Some easier ways of data augmentation are flipping, rotating, scaling, cropping, translating and adding gaussian or random noises to the images. Conditional GANs is a more advanced way where it can transform the images by the season it is in and even generating examples simply from the image datasets we have right now.

### **1.6 Report Organization**

The report is divided into 5 chapters. The literature review on transfer learning, recurrent neural network, action classification based on 3D convolutional network and review on related work are discussed in chapter 2. In chapter 3, the system design is discussed. Proceeding to chapter 4, this chapter described the experiments setup, analyze of results as well as evaluation of model and final product. Lastly, chapter 5 summarizes the project and discuss the future work.

## **Chapter 2 Literature Review**

### **2.1 CNN Transfer Learning**

Transfer learning is a popular technique by which a model that is trained and built can be reused when working on a similar task, instead of building and training a new model from scratch. This is very useful since most real-world issues typically do not have millions of datasets to train such complicated models. Few pre-train models will be discuss in this below section.

#### **2.1.1 Comparison between different pre-trained model**

In 2012, AlexNet was introduced and won the most challenging ImageNet competition called ImageNet Large Scale Visual Recognition Challenge (ILSVRC). It was a major advancement for visual recognition in the field of computer vision and machine learning. At the time, AlexNet was the first deeper and wider CNN that outperform in visual recognition. In the architecture of AlexNet, it contained eight layers where the first five layers are convolutional layers and the other three are fully connected layers. The first convolutional layer of AlexNet performs convolution and max pooling. The max pooling operations are performed using  $3 \times 3$  filters with a stride of 2. The similar operations are repeated in the second layer with filters of size  $(5 \times 5)$ .

In 2014, VGG-16 network was introduced by Simonyan and Zisserman (Simonyan & Zisserman, 2014). VGG-16 is a deeper neural network with small size of filter which is only  $(3 \times 3)$ . VGG-16 is a 16-layer architecture as indicate in the name which increase the number of layers from eight layers in AlexNet. Although currently Resnet architecture have more than 50 layers, VGG-16 network with 16-layer is still considered a very deep network in year 2014. At present, VGGNet has two variants of model which are VGG-16 and VGG-19. Since VGG apply small filters all the way, this result in VGG network have the same effective receptive field as if only have one convolutional layer of  $(7 \times 7)$ . The VGG architecture contains two convolutional layers that using ReLU activation function, followed by one max pooling layer and several fully connected layers that also implement ReLU activation function. SoftMax classifier is the final layer and responsible for classification. VGG network attained a low error rate of 6.8% and is runner up in ILSVRC in 2014. However, VGG is slower to train compared to other models in ILSVRC and the network architecture weights are heavy in terms of bandwidth (PyImageSearch, 2017).

Besides, GoogLeNet as one of the pretrain model was proposed by Christian Szegedy and it was designed to relieve the computation complexity of conventional CNN (Szegedy et al., 2015). GoogLeNet has an outstanding error rate at only 6.67% that made it to be the winner of ILSVRC 2014. GoogLeNet composed of 22 layers in its network architecture, which is greater than both the AlexNet and VGGNet. However, GoogLeNet has considerable low number of network parameter which is 7M as compared to the 60M of AlexNet and 138M of VGGNet. GoogLeNet enhance the up to minute recognition accuracy by implementing a stack of Inception layers as shown in Figure 2.1.1.1.

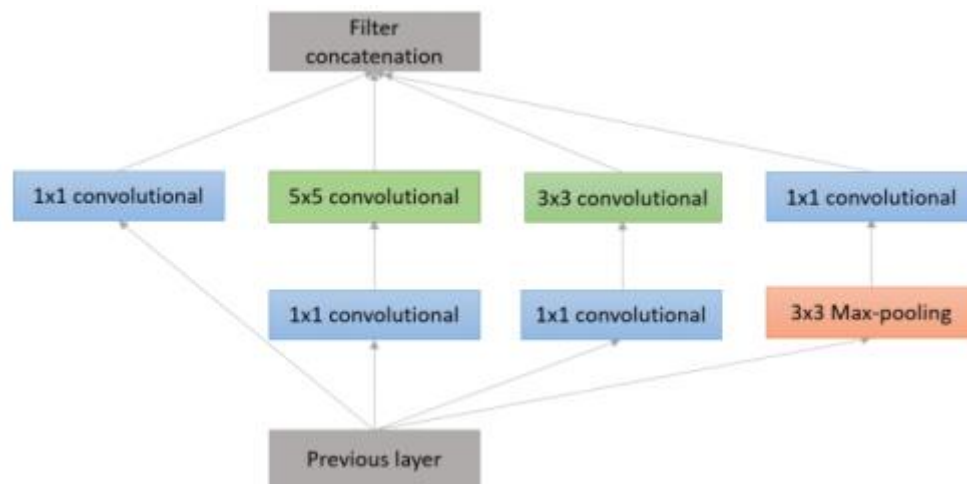


Figure 2.1.1.1: Inception Layers

Residual Neural Network (ResNet) was introduced by Kaiming during the ILSVEC of year 2015 (K.He et al., 2016). ResNet was a big breakthrough in the field of computer vision as ResNet do not have vanishing gradient issue that predecessors suffering. ResNet surpass all the predecessor AlexNet, VGGNet and GoogLeNet and become the winner of ILSVEC 2015. ResNet capable of training a massive number of layers as compared to the previous proposed pretrain model, while still having great performance using the concept of residual connection or skip connection. The working principle of the ResNet is introducing a residual connection that can skip more than one layers and allowed the network to extend to 152 layers while maintaining the lower complexity compared to VGGNet.

## 2.2 Action Recognition in Video Sequences using Deep Bi-Directional LSTM With CNN Features

Action recognition is split into two parts which are features extraction using pretrain CNN called AlexNet and then the features extracted are fed to the proposed Deep Bi-Directional LSTM. Bidirectional LSTM is not the same as the traditional LSTM, in which the output is depends on the previous data in sequence and also the upcoming data, while the normal LSTM is only depends on previous data in sequence. In order to implement bidirectional LSTM, two RNN is stacked on one another with one work in forward direction while another one in backward direction. The hidden state of both RNNs is then used to compute the combined output. Figure 2.2.1 shows the external structure of the training process, where the hidden states of forward and backward pass are merged in the output layer. After the output layer, the validation and cost are measured, as well as the weights and biases are modified using back-propagation. Figure 2.2.2 illustrates the internal structure of the bidirectional LSTM. This is a deep bidirectional LSTM since both forward and backward pass are made up of two LSTM cells. Since LSTM layers are processing in both directions, the output of a frame is determined from both the previous frame and the upcoming frame.

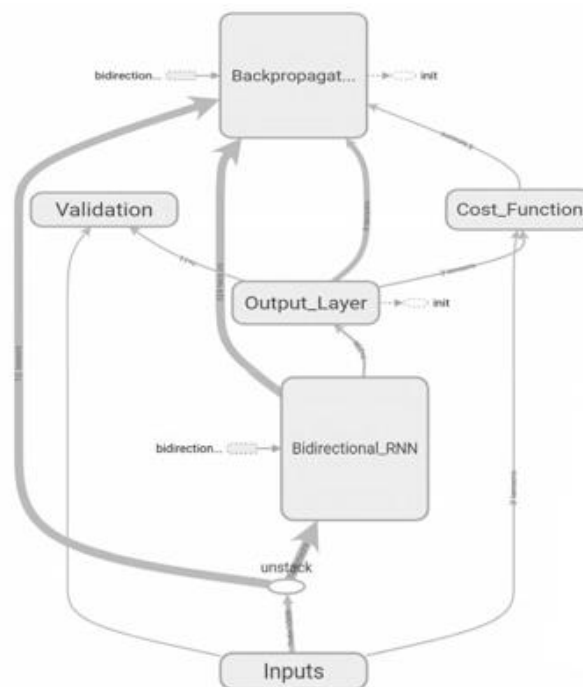


Figure 2.2.1: External Structure of proposed DB-LSTM network.

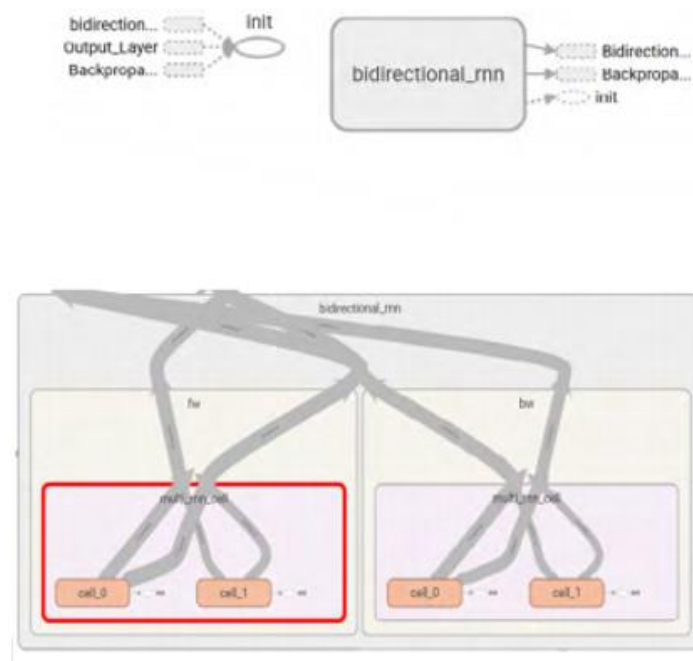


Figure 2.2.2: Internal Structure of proposed DB-LSTM network.

### 2.3 Action Classification Based on 3D Convolutional Networks

3D Convolution Network model is introduced to productively apply the temporal context information of the video sequence (Ji et al., 2013). With 3D convolution networks, the 2D convolution process can be generalized to 3D by stacking successive frames. 3D convolution kernel can be utilized to extract the temporal and spatial hierarchical feature in the video. Multiple 3D convolution kernels are stacked together on the identical convolutional layer to extract various features in the same spatial location. According to Karpathy et al., the slow fusion of 3D Convolutional Networks performs better than other temporal domain fusion approach (2014.). The time to train has also greatly reduced by constructing a multiresolution 3D Convolutional networks model.

Original 3D convolution kernel is proposed to be factorized to a 2D spatial convolution kernel and a 1D temporal kernels (Sun et al., 2015). 2D kernels are used by the lower layers to extract the video spatial features while upper network utilizes 1D convolution to examine the feature fusion on the time domain. The factorization process of the 3D convolution kernel not only reduces the time complexity of the model, but also reduces the requirement size of training data and prevent from overfitting.

Li et al. consolidate complex learning with C3D model and included a regularization constraint of spatio-temporal manifold in the loss function, successfully relieve the problem of overfitting and diminish variations in intra-class (2017). The 3D convolution operation has an invariance to the spatial transformation in the time sequence and has attained certain impacts. However, a time dimension channel added to the 2D convolution kernel by 3D convolutional kernel will significantly enlarge the number of parameters and computational cost.

## **2.4 Review on Related Work**

### **2.4.1 Intelligent human-centric lighting for mental wellbeing improvement**

In this paper, the author proposed an intelligent lighting system that able to improve the psychology health of human by changing different colour of light based on their current emotional state without manual intervention (Cupkova et al., 2019). Figure 2.4.1.1 depicts the system architecture of this lighting system. The main program logic is running on the Raspberry Pi which function as the central processing unit (CPU) while ESP8266 WIFI module is used to control LED strips through the radio frequency (RF). The steps in the main program is shown in Figure 2.4.1.2. First, the IP camera that is installed on the system responsible to capture the human emotion. When the videos are inputted into the system, the video will split into series of frames to detect human face and emotion. Based on the emotions that are recognized, the LED strips will be changed to suitable colour.

In this intelligent lighting system, the face detection algorithm proposed by Viola and Jones that applying the cascade classifiers concept is utilized to identify human faces in the video frames (Viola and Jones, 2004). Cascade classifier is trained to identify faces and provides immediate face detection. CNN is trained to classify human emotions. CNN model implements the concept of residual modules and depth-wise separable convolutions. Residual modules responsible in the mapping between 2 successive layers such that the learned features become the distinction between desired features and feature map. The desired features  $H(x)$  are adjusted to solve the learning problem  $F(x)$  such that

$$H(x) = F(x) + x$$

The advantage of residual modules is it permits gradient to be backpropagating more effectively in DNN. On the other hand, depth-wise separable convolutions can reduce the number of parameters by separating the processes of combination within convolutional layer and feature extraction.

The strength of this system is it applied artificial intelligence (AI) in recognizing emotions of humans and automatically calibrate light intensity as well as light colour. However, the weakness of this system is that the IP camera and Raspberry Pi 3 are connected through local network. The internet speed must be fast enough to transfer the sensor data from camera to CPU to process. This can be improved by connecting the camera to CPU to eliminate the transfer of data through network.

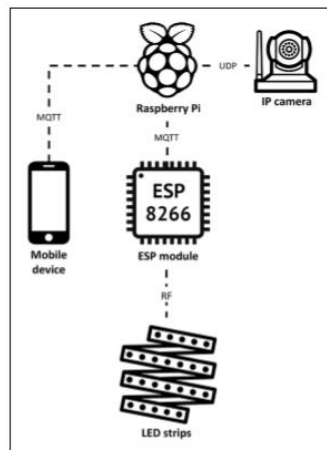


Figure 2.4.1.1 System scheme

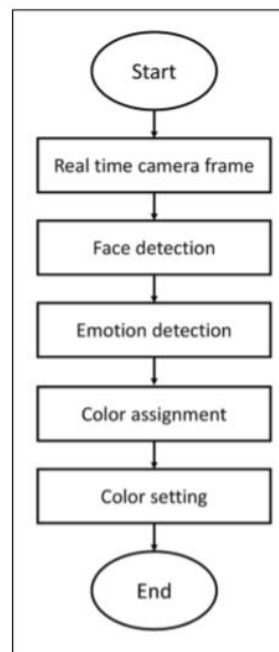


Figure 2.4.1.2 The flow of the system



## Chapter 3: Proposed Method/Approach

### 3.1 Methodology

To realize the project within projected timeframe, a research methodology has been proposed.

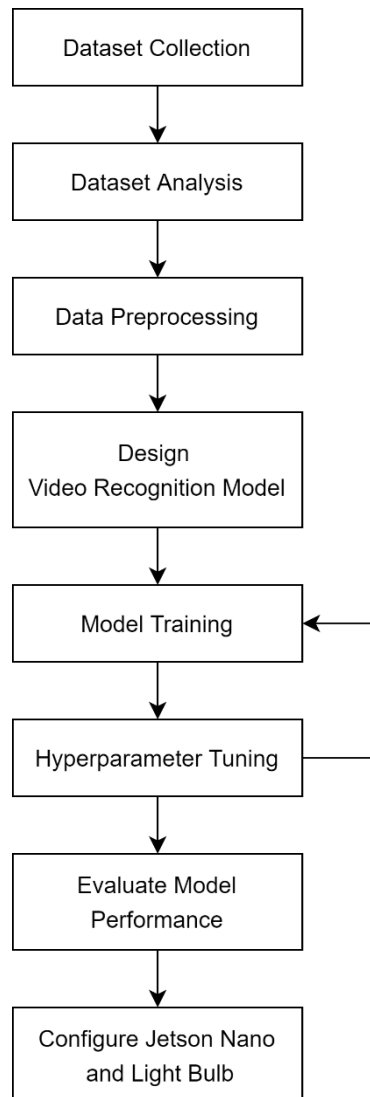


Figure 3.1.1: Research Methodology

In the first stage of the proposed project methodology, multiple standard datasets that are usually clean and available for public are collected from the Internet. During dataset analysis phase, multiple standard datasets that previously downloaded will be analyzed and only one dataset will be selected to be use in this project. After that, the classes of dataset to be used in this project will be determined and undergo screening process to filter out the low-quality video

clip in term of video resolution, video clarity and other aspects. Dataset labelling will be conducted and followed by splitting the datasets into training set, validation set, and testing set. Proceeding to data preprocessing stage, frame extraction of videos is conducted, followed by formatting and data transformation.

The next step of the proposed workflow is designing a video recognition model by referring to the schematics diagram from existing research paper. Once the architecture of the model is determined, the previously preprocessed training data and validation data are passed to the model for training. Validation data is used to measure the model performance at the end of each epoch. Two common issues which are underfitting and overfitting are usually happens in training phase. To deal with the issues, adjustments must be made such as hyperparameter tuning or adding dataset size to improve the model performance. Hyperparameter tuning involve the altering of epoch value, batch size, learning rate and etc.

The next phase after hyperparameter tuning will be model evaluation. Testing set that prepared earlier are used to test the model to understand the model generalization ability and see how it perform on the unseen data. In the final phase of the methodology, Jetson Nano and light bulb will be configured. The trained model will be installed on the Jetson Nano to perform the activity recognition and change the light temperature based on the action recognized.

### **3.2 Tools and technologies used**

In this project, Python programming language is selected to perform a set of complicated machines learning tasks. It provides access to libraries and frameworks such as Keras, TensorFlow and Scikit-learn for artificial intelligence (AI) and machine learning (ML). Keras and Tensorflow which are open-source neural network library are used to build the deep neural network. Besides, several Python libraries are also involved in this project such as pytube3 library is used for downloading the video datasets and ffmpeg library is used for trimming the video. Augmentation technique is taken from GitHub repository to perform horizontal flip on training video datasets. Lastly, OpenCV library is also involved to extract the frames from the videos to perform activity recognition.

### **3.3 Standard evaluation metrics for model performance**

Evaluation of the proposed deep learning model is essential to determine whether it achieves satisfactory result. To evaluate the effectiveness of a model, evaluation metrics such as accuracy, precision, recall and confusion matrix can be examined. Accuracy is one of most commonly used metric for evaluating how often the model in making a correct prediction. Accuracy is calculated by taking the number of correctly predicted observation divide the total observations. Unfortunately, accuracy is not a good indicator in determining the performance of model especially when the number of samples for each class are unequal. This is because when the testing set consist of 97% sample of class A and 3% of class B, the model can easily obtain 97% of accuracy by simply predicting all the sample belonging to class A correctly. However, when the testing set consist a large number of sample size from class B, the accuracy will be drop significantly due to the inability of predicting class B correctly. Thus, accuracy is a bad metric in evaluating model performance as it will give a false sense of obtaining high accuracy.

On the other hand, confusion matrix gives a more complete and deeper picture of how the model is performing. Confusion matrix depicts the ways in which the model is confused when making inference. Confusion matrix summarizes the number of correct and incorrect predictions for every classes and give an insight of which kinds of errors are being made. There are 4 important terms appeared in confusion matrix which are True Positive, True Negative, False Positive and False Negative that allows the computation of various classification metrics such as precision and recall.

Precision is also a popular metric in evaluating model performance. Precision attempt to answer the question, if the system predicts something to be true, how reliable is its prediction? Precision is calculated by taking the number of correctly predicted positive observations divided by the total number of predicted positive observations. From the formula, low precision essentially means that the model predicts a lot of false positives. Moreover, recall metric is defined as the amount of positive test samples that are predicted as positive. The high recall value of our classifier means that very few false negatives and that the classifier is more permissive in the criteria for classifying something as positive. One way to tell if the classifier is biased towards a positive class is if we have a very low precision, but a very high recall. Thus, the higher the precision and recall, the better the classifier performs because it detects most of the positive samples (high recall) and does not detect many samples that should not be detected (high precision).

### 3.4 Dataset Collection

This project made use of Moment in Time Dataset for the purpose of model training ([http://data.csail.mit.edu/soundnet/actions3/split2/Moments in Time Raw v2.zip](http://data.csail.mit.edu/soundnet/actions3/split2/Moments_in_Time_Raw_v2.zip)). Moment in Time Dataset is a collection of one million human-annotated short video that categorized to a total of 339 different classes. Each class comprises over 1000 videos of 3 seconds long which make it a massive and well-balanced dataset for learning dynamic events from video clips.

Since Moment in Time Dataset is one of the largest human-annotated video datasets, the total file size is approximately 275GB. The process of acquiring this dataset requires downloading the whole zip file because selective downloading is not supported. Once dataset is successfully downloaded, classes that will be used to train the deep learning model are determined. The next step after deciding the classes of dataset is screening and filtering out 100 videos that are better in resolution and clarity for each class. Equal size of sample for each class is crucial to avoid dataset imbalance issue that will cause the model bias toward one class during the training process.

The next step involves the labelling of dataset, follow by splitting dataset into training set, validation set and training set. The annotation file contains two attributes which represent directory and label for each video. The format of training annotation file and the directory structure of the processed dataset are as follow:

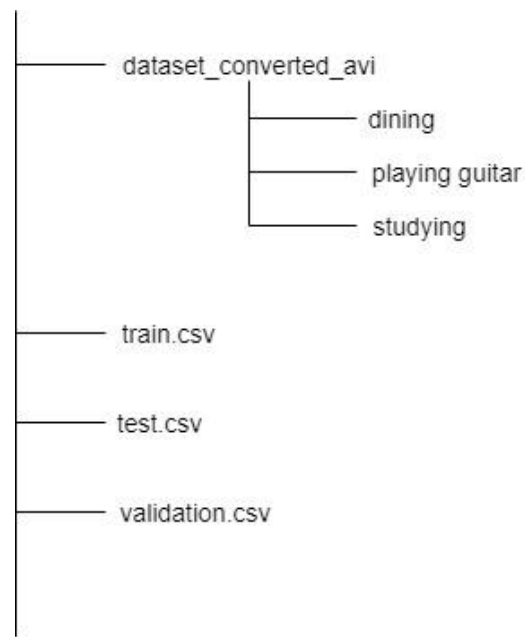


Figure 3.4.1: Directory Structure

Video	Label
./dataset_converted_avi/playing guitar/g91.avi	playing guitar
./dataset_converted_avi/dining/d80.avi	dining
./dataset_converted_avi/studying/s48.avi	studying
./dataset_converted_avi/studying/s76.avi	studying
./dataset_converted_avi/dining/d88.avi	dining
./dataset_converted_avi/studying/s90.avi	studying
./dataset_converted_avi/dining/d92.avi	dining
./dataset_converted_avi/playing guitar/g7.avi	playing guitar
./dataset_converted_avi/studying/s84.avi	studying
./dataset_converted_avi/playing guitar/g50.avi	playing guitar
./dataset_converted_avi/playing guitar/g29.avi	playing guitar
./dataset_converted_avi/studying/s71.avi	studying
./dataset_converted_avi/dining/d99.avi	dining
./dataset_converted_avi/studying/s95.avi	studying
./dataset_converted_avi/dining/d45.avi	dining
./dataset_converted_avi/dining/d100.avi	dining
./dataset_converted_avi/dining/d9.avi	dining
./dataset_converted_avi/dining/d33.avi	dining
./dataset_converted_avi/studying/s17.avi	studying

Figure 3.4.2: Training Annotation File

### 3.5 Data Pre-processing

The video clips obtained from Moment in Time Dataset need to be pre-processed to the desired format before can feed to the neural network for training. Firstly, all the video clips are extracted or break into series of RGB frames that consist of Red, Green and Blue (RGB) channels by using the library of OpenCV. The extracted frame number for each video is different, depending on the frame rate of the video. If the video is 3 seconds long and having frame rate of 30, the frame extraction process will produce 90 frames. After that, the extracted frame will be resized to appropriate image resolution and normalized to within 0 and 1. Since the pixel value of image data is in the range of 0 to 255, normalization is recommended to accelerate the training process because large values input can slow down the learning process. Normalization can be achieved by dividing all the pixels values by the maximum pixel value which is 255. Figure 3.5.1 shows the process of frames extraction.

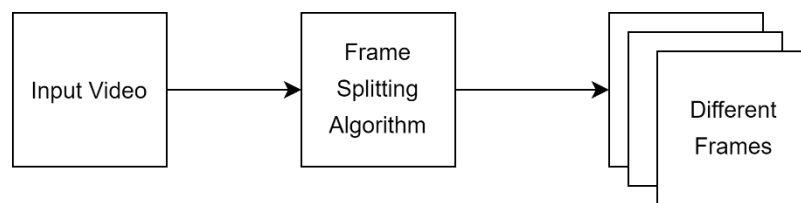


Figure 3.5.1: Frames Extraction Process

### 3.6 Proposed Network Architecture

Figure 3.5 shows the proposed CNN-LSTM framework. The proposed model consists of two main components which are the pretrain CNN model called VGG-16 that function as features extractor and LSTM for incorporating temporal information. The proposed CNN-LSTM works by passing each frame of a single video to the VGG-16 to extract the visual features. After feature extraction process, the feature map as the output from VGG-16 will be flattened to create a single long feature vector for inputting to the next layer of LSTM. Before passing the features to LSTM, a time series of data is required for LSTM to capture temporal information. This can be realized by concatenating the frames features obtained previously according to the frames sequence and then passes to LSTM layers for training and predicting the label of video.

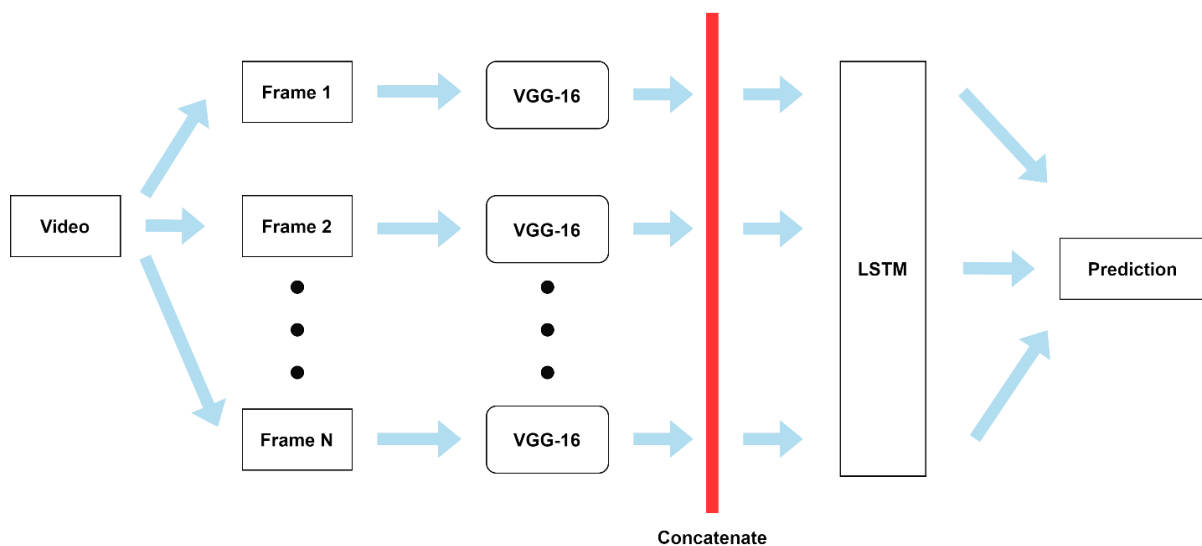


Figure 3.6.1: Proposed CNN-LSTM Network Architecture

#### 3.6.1 Fine-Tuned VGG-16 Architecture

VGG-16 had been selected as the pre-trained CNN model for feature extraction. Figure 3.6.1.1 illustrates the architecture of VGG-16. Feature extraction part of VGG-16 is beginning from the input layer to the last max pooling layer. To implement VGG-16 as the feature extractor, the final fully connected layers that used for classification are removed and the pre-trained convolutional layers are feezed. The reason of freezing the convolutional layers is to prevent the weight from updating because they are well trained to capture universal features. VGG-16 model is fine-tuned by adding a new max pooling layer with stride 2 at the end of feature

extraction part. The purpose of adding new max pooling layer is to reduce the dimension of the feature map to increase the training speed and decrease the memory consumption.

The default network's input is images of dimension (224, 224, 3). The unique of VGG-16 is that all convolutional layers in the network apply 3 x 3 filters with stride of 1 to the input. Throughout the network, there are many filters will be applied to the input to produce feature maps. For example, the first two convolutional layers comprised of 64 kernel filters and simple features such as edges, rotation and shapes are extracted by the neurons in these layers. The neurons learn to assemble the details to get a wider picture of the image in the following convolutional layers. After each set of convolutions layers, the feature maps output will undergo downsampling process in max pooling layer to reduce the number of parameters needed to train while maintaining the important features and makes the model more invariant to the minor distortion in input image. The convolution layer output size after passing through max pooling layer can be calculated by using the formula below:

$$\frac{1 - K + 2P}{S} + 1$$

where K, P and S denotes kernel size, padding size and stride respectively.

The first two convolutional layers in the network have the same padding and 64 channels of 3x3 filter size. When the input images are passed into these two layers, the dimension will change to (224,224,64). Then the output is passed to max pooling layer with 2x2 filter size and stride of 2 which reduce the dimension to (112,112,64). The third and fourth convolutional layers are comprised of 128 kernel filters followed by a max pooling layer. Proceeding to the fifth, sixth and seventh convolutional layers, all three apply 256 kernel filters and followed by another max pooling layer of stride 2. The last two sets of convolutional layers from eight to thirteen have 512 kernel filters. Each set of convolutional layers is followed by a max pooling layer of stride 2. Lastly, the new max pooling layer of stride 2 added to the end of feature extraction part will further reduce the feature map dimension and give the output dimension of (3, 3, 512). As compare to the output dimension (7, 7, 512) that yielded without a new max pooling layer added, the dimension of feature map now has effectively reduced by 5 times.



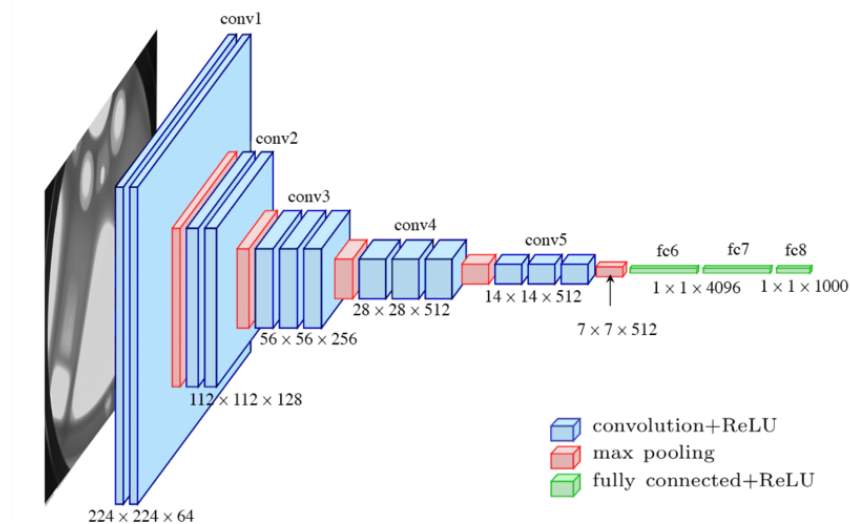


Figure 3.6.1.1: VGG-16 Architecture

### 3.6.2 LSTM Model Architecture

After all the features are extracted by the VGG-16, the output is reshaped or flattened into one dimensional array that fit the LSTM input layer shape. The input size for the LSTM model is  $20 \times 4608$  where 20 represent the time step of 20 and 4608 is obtained when feature maps  $(3, 3, 512)$  is reshaped to one dimensional array. Before passing the features maps to LSTM, a time series of data with time step of 20 is required to incorporate temporal information. Time step of 20 means the model will use the previous 20 frames to predict the label of next frame which exhibits the using of temporal information in making prediction. The time series data will then pass to the LSTM model for training. To avoid overfitting problem, only one LSTM layer is implemented followed by one SoftMax layer to perform classification. The architecture for the LSTM model implemented is shown in Figure 3.6.2.1.

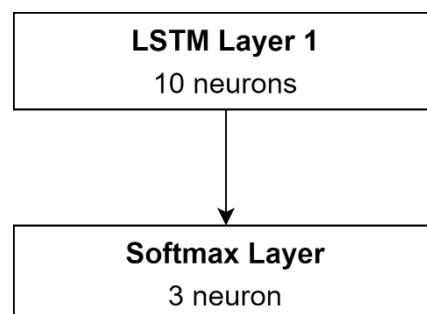


Figure 3.6.2.1: LSTM architecture

### 3.7 IoT Framework for Activity Recognition

The above activity recognition model has been extended to auto calibration of light colour temperature by implementing on Jetson Nano, which acts as the device for running inference locally. Figure 3.7.1 shows the auto calibration of colour temperature using embedded activity recognition. Following is the brief description of each step:

- i. Start the Jetson Nano
- ii. Initialized the embedded camera to captured human action
- iii. Send the captured frames to VGG-16 for feature extraction
- iv. Construct time series data and send to the trained LSTM model to perform recognition
- v. Change colour temperature of light bulb to the configured value based on each activity recognized

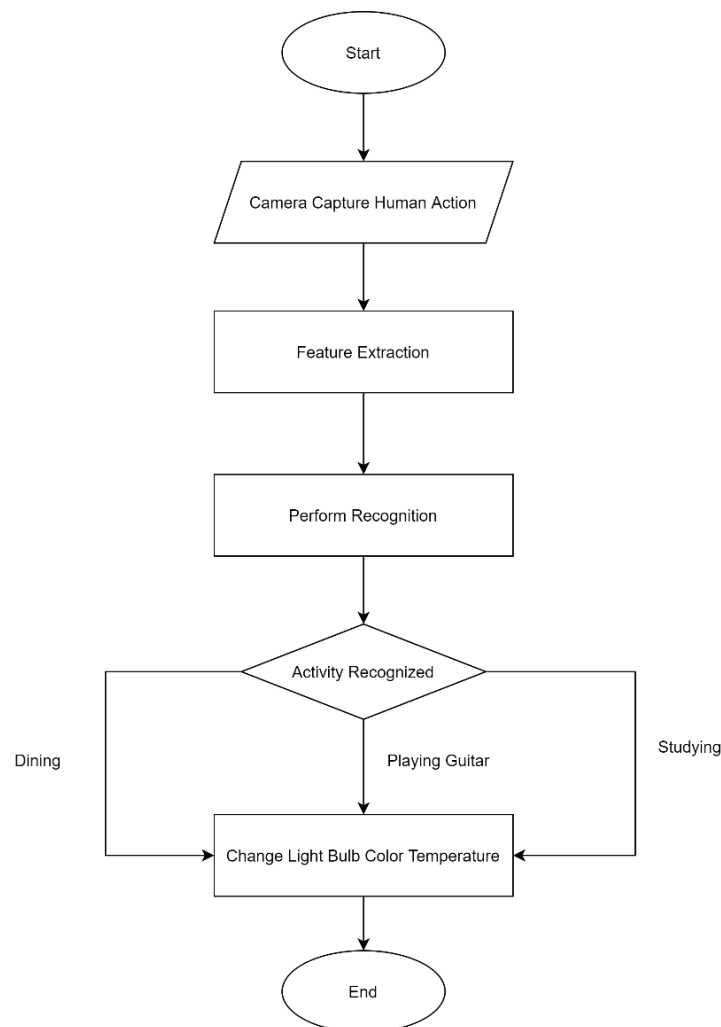


Figure 3.7.1: Flow Chart for Auto Tuning Light Colour Temperature

## Chapter 4: Experimental and Evaluation

### 4.1 Experiment Setup

In this experiment, dining, playing guitar and studying will be chosen from Moment in Time Dataset to train the model for activity recognition. The dataset was split into 70%, 15% and 15% for training, validation and testing set respectively. After splitting the datasets, the training videos are augmented using horizontal flip technique. The purpose of doing so is to double the training samples size. Performing data augmentation also create new videos with different orientation. The benefits of data augmentation are twofold, the first is to generate more data from limited dataset and secondly it helps to prevent overfitting issue.

Then, the training and validation videos are extracted into series of frames by passing the path of videos. The extracted frames are stored into train\_1 and validation\_1 folder. After extracting all the frames, the class name and the image name are written into csv. The frames are then normalized by taking the images to divide by 255 and loading them into NumPy array with the target size of (224,224,3) to fit the input shape of VGG-16.

The proposed CNN-LSTM network consists of two part which are using VGG-16 for features extraction and LSTM for model training. The architecture of the proposed network is defined. VGG-16 is used as the features extractor by removing the dense layer, freeze all the convolutional layers and adding a new max pooling layer with stride 2. The second part of the network made up of 1 LSTM layers and 1 dense layer for classification. The proposed network models will be trained on a workstation operating on Windows 10, equipped with Intel(R) Core(TM) i7-8550U processor and 16GB RAM.

Initially, the model is trained based on its default hyperparameter configurations as showed in Table 4.1.2. These hyperparameter might fine-tuned to improve the performance of the model during model training. The validation set is included during training to evaluate how well the model perform to new data.

The implementation details are depicted as below:

Python Library	Version
Keras	2.4.3
Tensorflow	2.3.0

Scikit-Image	0.14.2
OpenCV-python	4.4.0.42
yeelight	0.5.4

Table 4.1.1: Library Version

Parameter	Value
Batch size	128
Epoch	20
Optimizer	adam
Learning rate	0.001
Loss Function	categorical_crossentropy
Metric	accuracy

Table 4.1.2: LSTM Model Default Parameter

## 4.2 Jetson Nano Configuration

1. Prepare a microSD card and write the Jetson Nano Developer Kit SD Card Image to your microSD card by using a software called etcher (<https://www.balena.io/etcher>).
2. Insert the microSD card with system image into the slot of the Jetson Nano module.
3. Connect monitor display, mouse, keyboard, and camera module to Jetson Nano and power on it.
4. Complete the initial setting once the module is power on.
5. Install python 3.7 and other required libraries as shown in table 4.1.1.

## 4.3 Smart LED Bulb Configuration

The LED Bulb that used in this experiment is Yeelight Smart LED Bulb 1S. It supports the change of colour temperature which make it suitable to be used in this project. In order to establish a connection to Jetson Nano module over Wi-Fi, a YeeLight Python library is required to be installed (<https://gitlab.com/stavros/python-yeelight>). The features of this library are it allow the control of Yeelight LED bulbs over Wi-Fi such as changing color temperature, changing brightness, switch on and switch off.

#### 4.4 Baseline Model Evaluation

During the training process, the performance of the model is evaluated by using the validation data. To visualize how the model is performed, loss chart and accuracy chart are plotted for both training and validation. In the scenario where both training loss and validation loss are high, the model is subject to underfitting. Underfitting happens when the model cannot learn from the data feeding to it and hence perform poorly. On the other hand, when training loss and validation loss diverge, with low training loss, the model is suffering from overfitting. The training loss is low because the model learned too much from training data and cannot generalize to new data, which lead to high validation loss.

The default parameters are used for the baseline model during the initial training phase as shown in Table 4.1.2. and the model performance is illustrated in Figure 4.4.1. Accuracy-Loss Chart is plotted to understand how the model performed during the training process. From the figure, it can be seen that training loss decreased rapidly to approximately zero in first few epoch while validation loss did not show a notably decrease. Besides, the training accuracy is reaching one and validation accuracy recorded at 0.8. This shows a sign that the model might suffering from overfitting as it fit too well on training data. To overcome the overfitting issue, hyperparameter tuning can be performed to find the most suitable hyperparameter that give the best performance model.

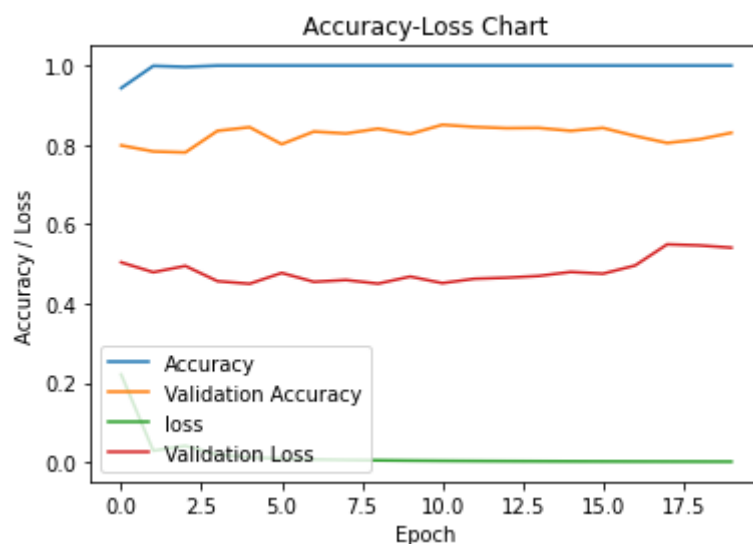


Figure 4.4.1: Accuracy-Loss Chart

### 4.5 Hyperparameter Optimization

This section discusses on determining the best configuration of hyperparameters for the LSTM models. Hyper-parameter tuning is the process of finding the best optimizing hyper-parameters for the model. Hyperparameters are those model's parameters that are not updated during the model training.

Table 4.5.1 below shows the selected values for each hyperparameter accompanied with the training result. The best performing model measured in term of validation loss is highlighted in green color.

	Learning Rate	Decay	Dropout	Early Stopping (Patience)	Batch Size	Epoch	Loss	Acc	Val Loss	Val Acc
1	1e-4	1e-5	0.5	2	128	5	0.0581	0.9999	0.4306	0.8294
2	1e-4	1e-5	0.5	2	64	4	0.0458	0.9989	0.4034	0.8526
3	1e-5	1e-4	0.5	2	128	39	0.0721	0.9992	0.3146	0.8832
4	1e-5	1e-4	0.5	2	64	9	0.1929	0.9717	0.6455	0.6893
5	1e-5	1e-5	0.5	2	128	30	0.0611	0.9996	0.2842	0.9064
6	1e-5	1e-5	0.5	2	64	26	0.0561	0.9993	0.3961	0.7696
7	1e-2	1e-5	0.5	2	128	3	0.0463	0.9976	0.2843	0.9211

Table 4.5.1: Result for different model parameters

Early stopping with patience of 2 is used during the experiment for the models to train until no further improvement in the next two consecutive epochs. Early stopping is a technique that enable an arbitrary large number of training epochs to be specified and stop training when the performance of model such as validation loss stops improving on the validation data. However, when patience of 2 is used, the model will only stop training when there is no further improvement on validation loss, and hence the model we will get is two epochs after the best model. To encounter this problem, model checkpoint callback can be used to save only the best model throughout the training process. The epochs showed in the table below is the epoch for the best model instead of full number of epochs the model has trained.

In this experiment, hyperparameters such as learning rate, decay rate, batch size and dropout are tweaked to find the best hyperparameter for the model. The model trained in the initial

training that based on the default parameters where the learning rate is 0.001 and dropout of 0 is used as the baseline model to verify if there is any improvement after hyperparameter tuning. Learning rate is a hyperparameter that regulate how often the weights are changed with respect to the loss gradient. A higher learning rate allows the model to converge rapidly and might cause undesirable divergent in loss function, while a lower learning rate will slow down the training process because very small updates are made to the network's weight. From Table 4.5.1, the learning rate of  $1e-2$  use the least epoch as compare to others to get to the optimal validation loss and after epoch 3, the loss function started to diverge. When loss function for learning rate of  $1e-4$  and  $1e-5$  are compared, the smaller learning rate will have a better divergence. So, setting the learning rate of  $1e-5$  is arguably better for model convergence.

Besides, dropout rate of 0.5 is introduced to the LSTM layer to prevent overfitting as occurred in the initial training. Instead of learning all the weights together, model now learn only 50% of weights in the network in each training iteration. Dropout rate is set to 0.5 throughout the hyperparameter tuning. Another technique to avoid overfitting is applying the weight decay by adding a small penalty to the loss function to keep the weights small. However, the weight decay cannot be too high as it will cause the model not fitting well. Based on Table 4.5.1, the validation loss for decay weight of  $1e-5$  is better than decay weight of  $1e-4$ .

Lastly, batch size is also one of the significant hyperparameters to tune. Small batch size will make the model converges faster at the cost of noise in the training process, while larger batch size offers a slower convergent learning process with accurate estimate of error gradient. From the result obtained in the table above, the batch size of 128 give the better performance with lower validation loss as compared to batch size of 64. This means that the larger batch size gives the better generalization ability in this project.

The Accuracy-Loss chart for best model is plotted at the end of hyperparameter optimization process as shown in Figure 4.5.1. The model after fined tune has better learning curve as compare to the model in initial training that using default parameters. The generalization gap between the validation loss and training loss is also reduced when the best model has better generalization on validation data.

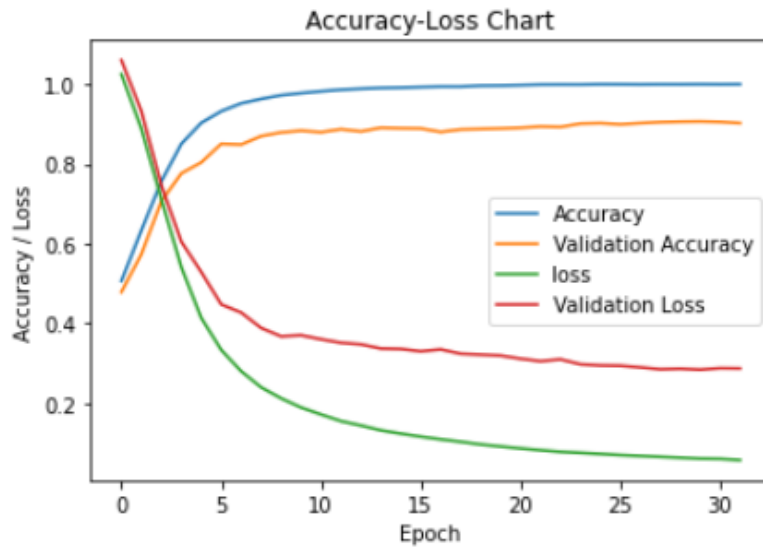


Figure 4.5.1: Accuracy-Loss Chart for Best Model

#### 4.6 Evaluation on Testing set

The testing sets that prepared earlier is used to provide an unbiased evaluation on the best model obtained in hyperparameter tuning phase and evaluation metrics such as accuracy, precision, recall and F1-score are examined.

The proposed model achieved a satisfactory accuracy of 75.94%. From the accuracy, we know that the model is doing a great job of recognizing three type of activities. However, accuracy can be misleading especially when dealing with imbalance dataset. A much better way of assessing the model performance is to gain an insight into the confusion matrix because it gives a deeper picture of how the model performs. Figure 4.6.1 depicts the confusion matrix for the model. The diagonal of the confusion matrix represents the number of samples that correctly classified by the model which is 1455. The model made a total of 461 error in performing prediction. Playing guitar recorded the highest number of errors in predicting the class label followed by studying. Dining has the least mistakes made by the model. From the data, we can see that the model is doing better in predicting dining than playing guitar and studying.

This proposed model also obtained a good precision value of 75.45%. Precision measure how reliable the model in predicting something to be true and thus it can be concluded that 75.45% of the instances classified in the positive class is correctly. Besides, the recall refers to the number of positive samples that were classified as positive. The recall of the model is 77.76% and it implies that 77.76% of the total numbers of positive instances were correctly classified.



Precision and recall can be combined into a single score known as F1 score. The proposed model obtained the F1 score of 75.865%. A high F1 score represents a low number of false positives and false negatives.

Precision, recall and F1 score for each class are also calculated and summarized in Table 4.6.2. Dining class has the highest recall value recorded at 89% among others. This means that the model able to classifies 89% of dining samples correctly. The highest precision of 0.80 is achieved by studying class and it implies that the model 80% reliable in classifying studying class correctly.

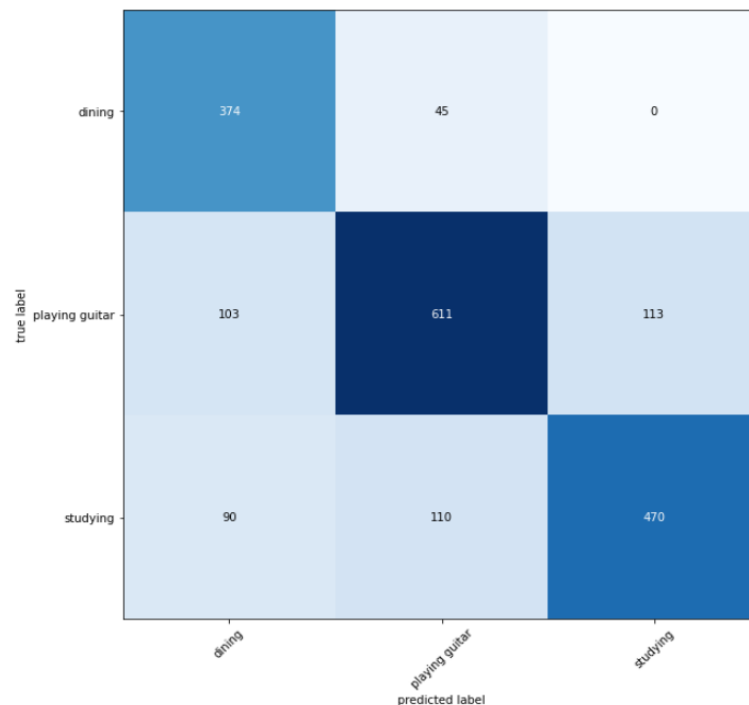


Figure 4.6.1: Confusion Matrix

Overall	Accuracy	Precision	Recall	F1 Score
	75.94%	75.45%	77.76%	75.87%

Table 4.6.1: Overall Performance of The Model

Class	Precision	Recall	F1 Score
<b>Dining</b>	0.6596120	0.8926014	0.7586207
<b>Playing Guitar</b>	0.7976501	0.7388150	0.7671061
<b>Studying</b>	0.8061750	0.7014925	0.7501995

Table 4.6.2: Performance metric for each class

#### 4.7 Implementation of Jetson Nano

The fine-tuned model is implemented on Jetson Nano to perform activity recognition and changing the colour temperature automatically. The model is tested by using the embedded camera to record human action and send for recognition. The result showed that the model leveraged on Jetson Nano able to perform correct prediction based on the recorded video. It is observed that the inference speed is slower compared to the workstation that used for training, which is probably due to the low computational capabilities.

Other than implementing the model on Jetson Nano, mapping of an ideal colour temperature of lightbulb to each activity recognized is also conducted. Colour temperature selected for each activity are based on past research regarding the most optimal colour temperature that can benefit every activity. The colour temperature ranging between 3,000-5,000K is recommended for home office or study area as it can help to concentrate with the tasks. On the other hand, dining room with warmer temperatures in the range of 2,200-3,000 K will bring cozy feeling. Since bedroom is the place for relaxing, the suggested color temperature should set at around 2,700K (Lighting, 2018). The playing of musical instrument is always in a relaxing mood and thus a warmer color temperature is recommended. Table 4.7.1 illustrated the color temperature value that will be map to each of the activity. Whenever the captured human action by the embedded camera on Jetson Nano is classified to any of these activities, the lightbulb will be calibrated to the optimal temperature that have been configured.

Activity	Optimal Colour Temperature
<b>Dining</b>	2700K
<b>Playing Guitar</b>	2500K
<b>Studying</b>	4500K

Table 4.7.1: Optimal Colour Temperature for each activity

#### 4.8 Final Product Evaluation

To verify whether the final product achieves the project objectives (iv), (v) and (vi), a survey is conducted to invite some experimenters to test on the product. There are a total of 5 experimenters that willing to participate in this survey. Each experimenter is given half an hour to test on the product. After trying on the product, they are requested to fill in the google form with 3 questions as shown in the figures below. The questions asked is related to the project objectives (iv), (v) ad (vi).

Based on the responses from the participants, all of them agreed that the smart lighting system able to improve the productivity in performing tasks, provide convenience and increase quality of living. However, there are two out of five responses show that the system is not benefits to health.

As a conclusion, it is believed that the final products achieved the project objectives (iv) and (vi) while the achievement of objectives (v) is vague as some of them agree that the system is beneficial to health while some of not disagree on it.

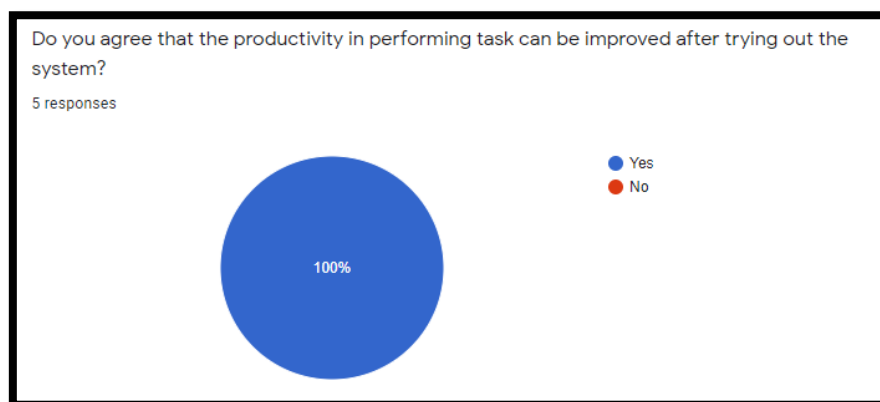


Figure 4.8.1: Data regarding of productivity

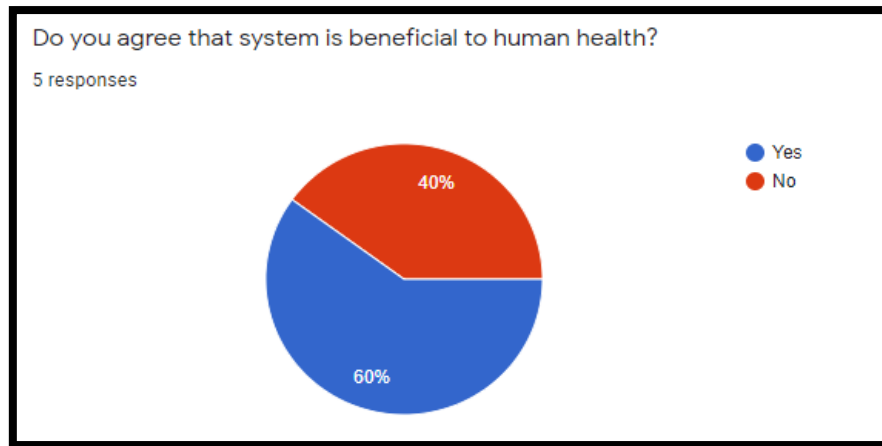


Figure 4.8.2: Data regarding of health

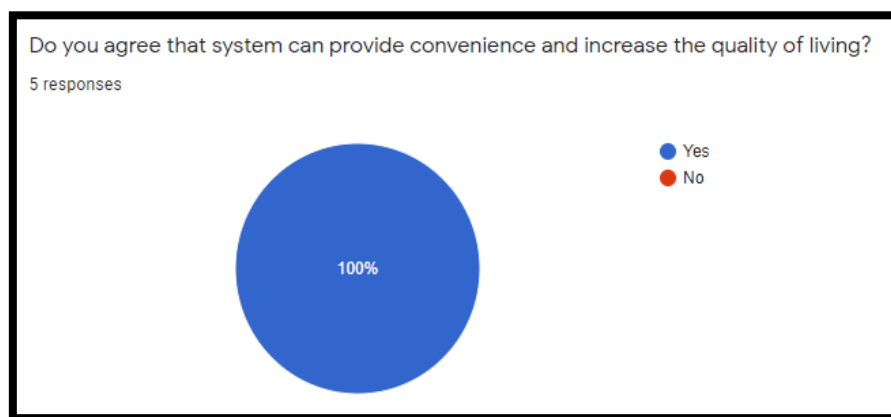


Figure 4.8.3: Data regarding of living quality

### **Chapter 5: Conclusion**

#### **5.1 Project Review**

In summary, light color temperature can have positive and negative impacts on human depends on the how and where to use them. The most notable effect is that working under incorrect light condition will damage visual acuity as well as decrease the productivity in doing daily tasks. Besides, even today there exist lights that can be tuned, it might cause troublesome to humans in calibrating the suitable color temperature with each different activity. Hence, this project aims to address the problems by designing a smart lighting system that able to auto calibrate the color temperature to the best suitable value when human activity is recognized through the camera sensor.

This project has proposed a CNN-LSTM model for the task of activity recognition. VGG-16 is implemented as a feature extractor to handle the spatial dependencies while LSTM network is implemented to capture temporal information and used as the classifier for activity recognition. Hyperparameter tuning of model is conducted to achieve a better performing model and successfully achieved a low validation loss of 0.2842 and high validation accuracy of 0.9064. The performance of model is evaluated on testing set and obtained a good accuracy of 75.94%, precision of 75.45%, recall of 77.76% and F1-score of 75.87%.

#### **5.2 Future Work**

As future work, a better technique for hyperparameter tuning such as AutoKeras may implemented for automatically finding the best-performing model instead of manually trying out different combination of parameters. AutoKeras will involve all the combination of parameters by passing a range to it. This will result in getting a better performing model.

Besides, more classes will be added to the dataset as currently only consists of 3 different type of activities due to time constraint and workstation computational capability issue. More classes implies that the model can recognize more type of activities and changing the optimal colour temperature based on the recognized action.

## BIBLIOGRAPHY

- He, K., Zhang, X., Ren, S. and Sun, J. (2016). *Deep Residual Learning for Image Recognition*. [online] Available at: [https://www.cv-foundation.org/openaccess/content\\_cvpr\\_2016/papers/He\\_Deep\\_Residual\\_Learning\\_CVPR\\_2016\\_paper.pdf](https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/He_Deep_Residual_Learning_CVPR_2016_paper.pdf).
- Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R. and Fei-Fei, L. (2014). *Large-scale Video Classification with Convolutional Neural Networks*. [online] Available at: <https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/42455.pdf>.
- Le, Q.V., Jaitly, N. and Hinton, G.E. (2015). A Simple Way to Initialize Recurrent Networks of Rectified Linear Units. *arXiv:1504.00941 [cs]*. [online] Available at: <https://arxiv.org/abs/1504.00941>.
- Li, C., Chen, C., Zhang, B., Ye, Q., Han, J. and Ji, R. (2017). Deep Spatio-temporal Manifold Network for Action Recognition. *arXiv:1705.03148 [cs]*. [online] Available at: <https://arxiv.org/abs/1705.03148> [Accessed 4 Sep. 2020].
- Lighting, I. (2018). *Lighting Inc / Tulsa Lighting Supply*. [online] Lighting, Inc. Available at: [http://www.lightinginc.us/blog/blog\\_posts/view/1/%E2%80%9Cwhat-is-the-proper-led-color-temperature-for-this-room%E2%80%9D](http://www.lightinginc.us/blog/blog_posts/view/1/%E2%80%9Cwhat-is-the-proper-led-color-temperature-for-this-room%E2%80%9D) [Accessed 15 Apr. 2021].
- Mills, P., Tomkins, S. and Schlangen, L., 2007. The effect of high correlated colour temperature office lighting on employee wellbeing and work performance. *Journal of Circadian Rhythms*, 5(0), p.2.
- PyImageSearch. (2017). *ImageNet: VGGNet, ResNet, Inception, and Xception with Keras*. [online] Available at: <https://www.pyimagesearch.com/2017/03/20/imagenet-vggnet-resnet-inception-xception-keras/>.
- Shamsul, B. M. T., Sia, C. C., Ng, Y. G., & Karmegan, K. (2013). Effects of light's colour temperatures on visual comfort level, task performances, and alertness among students. *American Journal of Public Health Research*, 1(7), 159-165.


## BIBLIOGRAPHY

- Simonyan, K. and Zisserman, A. (2014). *Very Deep Convolutional Networks for Large-Scale Image Recognition*. [online] ResearchGate. Available at: [https://www.researchgate.net/publication/265385906\\_Very\\_Deep\\_Convolutional\\_Networks\\_for\\_Large-Scale\\_Image\\_Recognition](https://www.researchgate.net/publication/265385906_Very_Deep_Convolutional_Networks_for_Large-Scale_Image_Recognition) [Accessed 3 Sep. 2020].
- Sun, L., Jia, K., Yeung, D.-Y. and Shi, B.E. (2015). Human Action Recognition using Factorized Spatio-Temporal Convolutional Networks. *arXiv:1510.00562 [cs]*. [online] Available at: <https://arxiv.org/abs/1510.00562> [Accessed 4 Sep. 2020].
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D. and Vanhoucke, V. (2015). Going deeper with convolutions. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [online] Available at: <https://www.cs.unc.edu/~wliu/papers/GoogLeNet.pdf> [Accessed 22 Jan. 2019].
- Vinyals, O., Ravuri, S.V. and Povey, D. (2012). Revisiting Recurrent Neural Networks for robust ASR. *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*
- Viola, P. and Jones, M.J. (2004). Robust Real-Time Face Detection. *International Journal of Computer Vision*, [online] 57(2), pp.137–154. Available at: <http://www.vision.caltech.edu/html-files/EE148-2005-Spring/pprs/viola04ijcv.pdf>.

## APPENDIX A: Poster


# PERVASIVE IOT: AUTO CALIBRATION IN LIGHT SENSOR USING EMBEDDED ACTIVITY RECOGNITION

Present by Neow Zhi Chin




## INTRODUCTION

An incorrect application of colour temperature will lead to loss of productivity, damage vision ability and affect the well-beings. This project aims to maximize the productivity of human in doing daily task, improved health and protect the eye vision ability while replacing the conventional way of tuning the colour temperature to autonomous




## OBJECTIVES




- To build an autonomous smart lighting system that can change colour temperature based on human activity with embedded activity recognition.
- To eliminate delayed inference by running inference computations locally on the device without sending request and waiting cloud-based service to response.
- To evaluate the accuracy of activity recognition.
- To enhance the human productivity in performing daily task by creating optimum lighting condition.
- To provide automatic color temperature tunable lighting designed to improve human health and wellness.
- To provide a level of comfort and convenience as well as increase quality of living.

## METHODOLOGY

CNN-LSTM model that combining VGG-16 as a feature extractor and LSTM as classifier is proposed to perform activity recognition



## RESULTS



The experimental results show that the proposed model behave well and achieves 75.45 % of precision and 77.76% of recall.

## CONCLUSION

This project is designing a smart lighting system that able to auto calibrate the color temperature to the best suitable value when human activity is recognized through the camera sensor. The motivation of this work is that correct application of color temperature can increase productivity in performing daily tasks, enhance health as well as beneficial to visual acuity of humans and at the same moment, it can improve the living quality of humans.



**APPENDIX B: Final Year Project Weekly Report****FINAL YEAR PROJECT WEEKLY REPORT***(Project II)*

<b>Trimester, Year: Y3S3</b>	<b>Study week no.: 2</b>
<b>Student Name &amp; ID: Neow Zhi Chin 18ACB04706</b>	
<b>Supervisor: Dr. Aun Yichiet</b>	
<b>Project Title: PERVASIVE IOT: AUTO CALIBRATION IN LIGHT SENSOR USING EMBEDDED ACTIVITY RECOGNITION</b>	

<b>1. WORK DONE</b> [Please write the details of the work done in the last fortnight.]  -Meet with supervisor to discuss how to start the project
<b>2. WORK TO BE DONE</b>  i. Understand the details of overall project
<b>3. PROBLEMS ENCOUNTERED</b>  No
<b>4. SELF EVALUATION OF THE PROGRESS</b>  Self-assigned tasks can be finished within timeframe



Supervisor's signature



Student's signature

## FINAL YEAR PROJECT WEEKLY REPORT

*(Project II)*

<b>Trimester, Year: Y3S3</b>	<b>Study week no.: 4</b>
<b>Student Name &amp; ID: Neow Zhi Chin 18ACB04706</b>	
<b>Supervisor: Dr. Aun Yichiet</b>	
<b>Project Title: PERVASIVE IOT: AUTO CALIBRATION IN LIGHT SENSOR USING EMBEDDED ACTIVITY RECOGNITION</b>	

<b>1. WORK DONE</b> [Please write the details of the work done in the last fortnight.]  -Finding new video dataset
<b>2. WORK TO BE DONE</b> Planning to collect new dataset
<b>3. PROBLEMS ENCOUNTERED</b>  Lack of good datasets
<b>4. SELF EVALUATION OF THE PROGRESS</b> Satisfy with the progress



\_\_\_\_\_  
Supervisor's signature



\_\_\_\_\_  
Student's signature

# FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

<b>Trimester, Year: Y3S3</b>	<b>Study week no.: 6</b>
<b>Student Name &amp; ID: Neow Zhi Chin 18ACB04706</b>	
<b>Supervisor: Dr. Aun YiChiet</b>	
<b>Project Title: PERVASIVE IOT: AUTO CALIBRATION IN LIGHT SENSOR USING EMBEDDED ACTIVITY RECOGNITION</b>	

<b>1. WORK DONE</b>  Dataset downloaded
<b>2. WORK TO BE DONE</b>  Explore video recognition model
<b>3. PROBLEMS ENCOUNTERED</b>  No
<b>4. SELF EVALUATION OF THE PROGRESS</b>  Progress on Track



Supervisor's signature



Student's signature

# FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

<b>Trimester, Year: Y3S3</b>	<b>Study week no.: 8</b>
<b>Student Name &amp; ID: Neow Zhi Chin 18ACB04706</b>	
<b>Supervisor: Dr. Aun YiChiet</b>	
<b>Project Title: PERVASIVE IOT: AUTO CALIBRATION IN LIGHT SENSOR USING EMBEDDED ACTIVITY RECOGNITION</b>	

<b>1. WORK DONE</b> RNN Model has been built successfully
<b>2. WORK TO BE DONE</b> Train Model
<b>3. PROBLEMS ENCOUNTERED</b> Input shape of RNN is difficult to do
<b>4. SELF EVALUATION OF THE PROGRESS</b> Progress on Track



Supervisor's signature



Student's signature

# FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

<b>Trimester, Year: Y3S3</b>	<b>Study week no.: 10</b>
<b>Student Name &amp; ID: Neow Zhi Chin 18ACB04706</b>	
<b>Supervisor: Dr. Aun YiChiet</b>	
<b>Project Title: PERVASIVE IOT: AUTO CALIBRATION IN LIGHT SENSOR USING EMBEDDED ACTIVITY RECOGNITION</b>	

## 1. WORK DONE

Model has been trained

## 2. WORK TO BE DONE

- Improve performance of model
- Adding VGG-16 to extract features
- Fine Tuning of Model

## 3. PROBLEMS ENCOUNTERED

Model overfitting

## 4. SELF EVALUATION OF THE PROGRESS

Progress on Track



Supervisor's signature



Student's signature

# FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

<b>Trimester, Year: Y3S3</b>	<b>Study week no.: 12</b>
<b>Student Name &amp; ID: Neow Zhi Chin 18ACB04706</b>	
<b>Supervisor: Dr. Aun YiChiet</b>	
<b>Project Title: PERVASIVE IOT: AUTO CALIBRATION IN LIGHT SENSOR USING EMBEDDED ACTIVITY RECOGNITION</b>	

<b>1. WORK DONE</b>  Model performance improved
<b>2. WORK TO BE DONE</b>  Build Jetson Nano with trained model and configure smart light
<b>3. PROBLEMS ENCOUNTERED</b>  Jetson Nano RAM too small, easily out of memory
<b>4. SELF EVALUATION OF THE PROGRESS</b>  Progress on Track



Supervisor's signature



Student's signature

APPENDIX C: Plagiarism Check Result

Document Viewer

Turnitin Originality Report

Processed on: 16-Apr-2021 04:48 +08  
ID: 1359593638  
Word Count: 7932  
Submitted: 2  
FYP2 By Neow Zhi Chin

include quotedinclude bibliographyexcluding matches < 8 wordsmode: quickview (classic) reportchange modeprintdownload

Similarity Index  
11%

Similarity by Source  
Internet Sources: 6%  
Publications: 7%  
Student Papers: 4%

2% match (publications) "Data Science", Springer Science and Business Media LLC, 2019
1% match (Internet from 24-Nov-2020) <a href="https://www.ibm.com/cloud/learn/recurrent-neural-networks">https://www.ibm.com/cloud/learn/recurrent-neural-networks</a>
1% match (Internet from 17-Feb-2020) <a href="http://kee.lib.auth.gr">http://kee.lib.auth.gr</a>
1% match (Internet from 01-Apr-2021) <a href="https://simonhessner.de/why-are-precision-recall-and-f1-score-equal-when-using-micro-averaging-in-a-multi-class-problem/">https://simonhessner.de/why-are-precision-recall-and-f1-score-equal-when-using-micro-averaging-in-a-multi-class-problem/</a>
1% match (Internet from 10-Sep-2020) <a href="https://journals.sagepub.com/doi/10.1177/1550147719875878">https://journals.sagepub.com/doi/10.1177/1550147719875878</a>
1% match (Internet from 28-Mar-2021) <a href="https://www.mdpi.com/2079-9292/8/3/292/htm">https://www.mdpi.com/2079-9292/8/3/292/htm</a>
<1% match (student papers from 11-Sep-2017) Submitted to Grand Canyon University on 2017-09-11
<1% match (publications) Zirui Qiu, Jun Sun, Mingyue Guo, Mantao Wang, Dejun Zhang, "Chapter 1 Survey on Deep Learning for Human Action Recognition", Springer Science and Business Media LLC, 2019
<1% match (publications) Mathew Monfort, Carl Vondrick, Aude Oliva, Alex Andonian et al, "Moments in Time Dataset: One Million Videos for Event Understanding", IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020
<1% match (student papers from 24-Jul-2020) Submitted to National Institute of Technology, Silchar on 2020-07-24
<1% match (Internet from 12-Jan-2021) <a href="https://pure.uva.nl/ws/files/54266896/1_s2.0_S2589871X20300176_main.pdf">https://pure.uva.nl/ws/files/54266896/1_s2.0_S2589871X20300176_main.pdf</a>
<1% match (student papers from 31-Oct-2016) Submitted to Aberystwyth University on 2016-10-31
<1% match () <a href="http://arxiv.org">http://arxiv.org</a>
<1% match (student papers from 20-Nov-2020) Submitted to NCG on 2020-11-20

feedback studio

Neow Zhi Chin | FYP2

Match Overview

11%

11

1 "Data Science", Springe... Publication 2%

2 www.ibm.com Internet Source 1%

3 ikee.lib.auth.gr Internet Source 1%

4 simonhessner.de Internet Source 1%

5 journals.sagepub.com Internet Source 1%

6 www.mdpi.com Internet Source 1%

7 Submitted to Grand Ca... Student Paper <1%

8 arxiv.org Internet Source <1%

9 Zirui Qiu, Jun Sun, Min... Publication <1%

10 Mathew Monfort, Carl ... Publication <1%

11 Submitted to National I... Student Paper <1%

12 Submitted to Aberystw... Student Paper <1%

13 pure.uva.nl <1%

Text-only Report High Resolution On

Page: 1 of 23 Word Count: 7932

The Internet of Things (IoT) is a hot issue nowadays and the rapid developed of the IoT has promoted smart lighting coming into our lives. A lot of ways have been introduced in controlling the lights such as remotely adjust the light brightness. Nevertheless, until today, smart lighting system that capable of adjusting colour temperature automatically based on human activities without requires manual intervention is still unknown. Since there is no existing method that can change the colour temperature automatically based on the current activity, this gave birth to the new idea of developing a smart lighting system using embedded activity recognition. The motivation of this work is that the optimal colour temperature is significant and beneficial to health, productivity and visual acuity of humans. The correct application of colour temperature in an indoor environment can increases productivity, supports cognitive processes and enhance health.

Besides, in the area of activity recognition, most of the training and inference in these days happens in the cloud. There exists severe disadvantage to this approach which is the delay of inference processing during a network congestion or when the cloud servers are overloaded. To cope with this issue, performing deep inference on local device is a better solution. In the local components, the recognition



## APPENDIX D – TURNITIN FORM

<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>	NEOW ZHI CHIN
<b>ID Number(s)</b>	18ACB04706
<b>Programme / Course</b>	CS
<b>Title of Final Year Project</b>	PERVASIVE IOT: AUTO CALIBRATION IN LIGHT SENSOR USING EMBEDDED ACTIVITY RECOGNITION

<b>Similarity</b>	<b>Supervisor's Comments</b> (Compulsory if parameters of originality exceeds the limits approved by UTAR)
<b>Overall similarity index:</b> <u>11</u> % <b>Similarity by source</b> Internet Sources: <u>6</u> % Publications: <u>7</u> % Student Papers: <u>4</u> %	
<b>Number of individual sources listed</b> 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: Dr. Aun Yichiet  
Date: 15/04/2021

Signature of Co-Supervisor  
Name:  
Date:

## APPENDIX E – CHECKLIST





**UNIVERSITI TUNKU ABDUL RAHMAN**  
**FACULTY OF INFORMATION & COMMUNICATION**  
**TECHNOLOGY (KAMPAR CAMPUS)**  
**CHECKLIST FOR FYP2 THESIS SUBMISSION**

Student Id	18ACB04706
Student Name	Neow Zhi Chin
Supervisor Name	Dr. Aun Yichiet

<b>TICK (√)</b>	<b>DOCUMENT ITEMS</b> 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)

I, the author, have checked and confirmed all the items listed in the table are included in my report.   _____ (Signature of Student) Date: 15/04/2021	Supervisor verification. Report with incorrect format can get 5 mark (1 grade) reduction.   _____ (Signature of Supervisor) Date: 15/04/2021
--	---