MONASH University

Department of Electrical &
Computer Systems Engineering

Automated Video-based Epilepsy Detection and Classification using Deep Learning

Zhaoning Lu

Student ID: 29386993

Supervisor: Dr. Zongyuan Ge

Significant Contributions

- Designed and implemented a pipeline for extracting joint-semiology and motion features which are now (and in the future) deployed at the Alfred Hospital for collecting privacy-preserving features from the video data of seizure patients

- Implemented state-of-the-art techniques based on motion features for automated detection of epileptic seizures on pilot data from Alfred Hospital

MONASH University Engineering

Department of Electrical and Computer Systems Engineering

ECE4095 Final Year Project 2022

Zhaoning Lu

Automated Video-based Epilepsy Detection and Classification using Deep Learning

Supervisor: Dr. Zongyuan Ge

Project Background & Aim

- Epilepsy is a periodical repeated seizure attack usually caused by disordered brain electrical rhythms. This kind of turbulence in bio-electricity signal will result in uncontrolled limbs and facial expressions and even cause absence and fatal consequences. From years of medical studies, it has been recognized that epilepsy is uncorrelated to gender, race, or age. And according to the WHO [1], more than 50 million people suffer from epileptic seizures worldwide; it is one of the most common global diseases.
- In this project, we focused on two types of seizures: 1. The Generalized Tonic-Clonic Seizures (GTCS), and 2. The Psychogenic Nonepileptic Seizures (PNES). As the name suggests, the PNES is not an epileptic symptom; instead, it is a paroxysmal symptom. Even though the patient's outer behavior for the two types of seizures is close, the medical treatments are entirely different. Based on statistical results, about 25 % of PNES patients have been misdiagnosed with GTCS [2]. The traditional way applied in clinical environments relies on EEG, but there are cases where patients cannot tolerate wearable devices.

This project aims to find a way to detect seizures using Deep Learning. There are two types of data modalities have been used to train the networks: optical flow, and joint-semiology.

Overview

Data extraction and storing

During the project, the Alfred Hospital has provided us with video data of seizures; considering privacy issues, all the dataset we generated and collected is privacy-preserved features. The feature extraction is done on Nvidia Jetson Xavier, which is now (and in the future) deployed at Alfred Hospital to generate more data samples.

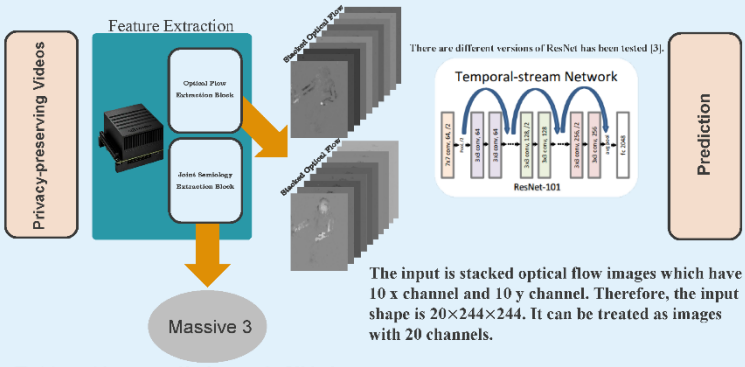
Network Validation

Two models have been implemented to test the performance of the optical-flow-based and joint-semiology-based action recognition and classification, which are 'two-stream-action-recognition' and 'HCN-pytorch'.

Pilot Test

Implemented state-of-the-art techniques for automated detection of epileptic seizures on pilot data from Alfred Hospital.

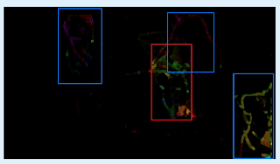
Methods



The input is stacked optical flow images which have 10 x channel and 10 y channel. Therefore, the input shape is 20×244×244. It can be treated as images with 20 channels.

The joint-semiology extracted has been saved to M3 for future use.

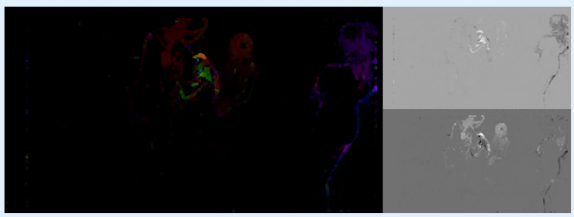
Challenges



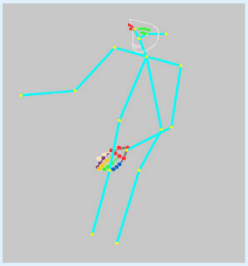
- One common issue for optical-flow-based and joint-semiology-based epilepsy recognition is that the medical staff would block the patient frequently. As shown in the figure, the red box is the patient's location, and the blue box is the medical staff's location. (ROI mechanism has been used to improve this issue.)
- Another difficulty is that many videos were recorded during the night using infrared cameras, making the feature extraction becomes unstable.

Results & Future Improvements

- The figure below shows the extracted optical-flow features of patients. The image in Grayscale is the u and v channel separated from the RGB channel.



- The outcome of integrated human-joint semiology is shown below. The semiology for the body is based on the 'light-weight-human-pose-estimation' method, and the facial and hand semiology is based on 'MediaPipe'.



- The training accuracy for optical-flow-based epilepsy recognition is 74.74%, and the testing accuracy is 65.52%.

Conditions	Training Accuracy	Testing Accuracy	With ROI	With Further Split
Res18-Stack Size 10	73.199%	63.727	Y	Y
Res18-Stack Size 20	73.346%	64.826%	Y	Y
Res18-Stack Size 50	69.778%	58.874%	Y	Y
Res34-Stack Size 10	73.170%	62.003%	Y	Y
Res34-Stack Size 20	74.741%	65.517%	Y	Y
Res34-Stack Size 50	67.455%	55.509%	Y	Y
Res101-Stack Size 10	71.829%	61.201%	Y	Y
Res101-Stack Size 20	70.532%	62.650%	Y	Y
Res101-Stack Size 50	68.114%	55.780%	Y	Y

- At the current stage, dataset insufficiency (only 9 GTCS videos have been provided) is the primary constraint for gaining a high prediction accuracy. The feature extraction process will keep operating in the Alfred Hospital to generate more data.

[1] "Epilepsy", *Who.int*, 2022. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/epilepsy.
[2] "The Truth about Psychogenic Nonepileptic Seizures," Epilepsy Foundation. [Online]. Available: https://www.epilepsy.com/stories/truth-about-psychogenic-nonepileptic-seizures
[3] Yi Huang, Rajat Shrivastava "two-stream-action-recognition: Using two stream architecture to implement a classic action recognition method on UCF101 dataset", *GitHub*, 2022. [Online]. Available: https://github.com/jeffreyyihuang/two-stream-action-recognition.

Executive Summary

This report starts with the background of epilepsy seizures and the motivations for developing a way to detect and classify epilepsy seizures based on deep learning. The design and implantation process has been detailed, including dataset extraction and storing, deep learning model validation, and pilot test.
The model used to perform the epilepsy seizures detection is based on Residual Network; a significant improvement in accuracy has been observed after applying the ROI and dataset further split mechanism. Several comparison experiments were also listed at the end of the report.

Acknowledgements

Thanks to the Monash Medical AI group, they have given me this opportunity to dig into this meaningful topic related to human health. And thanks to the Alfred Hospital for providing epilepsy videos, which made this project possible.
A special thanks to Dr. Deval Mehta, who has patiently helped me tackle many problems. And motivated me to keep pushing myself.

Contents

1. Introduction

1.1 Epileptic Seizures

The epileptic seizure, also known as epilepsy, is a periodical repeated seizure attack usually caused by brain disordering/abnormal. From the observation of medical instruments like EEG (electroencephalogram), the brain's electrical rhythms become disordered and imbalanced for patients under symptom onset. This kind of turbulence in bio-electricity signal will result in uncontrolled limbs and facial expressions and even cause absence and fatal consequences. From years of medical studies, it has been recognised that epilepsy is uncorrelated to gender, race, or age. This means anyone would have the possibility of developing epileptic seizures. The medical treatment varies from person to person, some people would need lifelong treatment, but the symptom will disappear for some people as they age. As data provided by the World Health Organization [1], more than 50 million people are suffering from epileptic seizures worldwide; it is one of the most commonly global diseases. And inside these groups of people, about 80% of them are not living in a high-income country, which means the medical treatments they can get are highly likely in a shortage. More importantly, WHO estimated that 70% of these people could cure this disease if they can be diagnosed correctly and applied with appropriate medical treatment. Unfortunately, most of them still cannot get proper medical treatment. There are many causes of this disease; most commonly, the reason will be loss of oxygen, congenital brain conditions, head injury, or brain infections. In fact, there are many different types of epilepsy, and the treatment for different kinds of them is not the same. Using incorrect therapy for different types of epileptic attacks is less meaningful. In this project, the focus of the epileptic seizures is the Generalized Tonic-Clonic Seizures (GTCS), which have very close symptoms to the psychogenic nonepileptic seizure.

1.2 Psychogenic Nonepileptic Seizures (PNES)

Seizures can be split into two categories: 1. The epileptic seizures discussed in the previous section, and 2. The psychogenic nonepileptic seizures. As the name suggests, the PNES is not an epileptic symptom but is a paroxysmal symptom. However, the patient's behaviour for a PNES attack is very close to an epileptic attack. Instead of being caused by disordering in brain bio-electrical effect, psychogenic nonepileptic seizures are psychological distress. Thus, the medical treatment for this kind of seizure is way different from the GTCS. From the research based on [2], about 25 per cent of PNES patients have been misdiagnosed with GTCS. The result is that they do not respond to their wrong treatment and thereby waste time and money. Traditionally, the most reliable diagnostic method for PNES is usually based on EEGs.

1.3 The Alfred Hospital and Monash Medical AI Group

Due to the high demand for real patient data (videorecording) for developing the model, the project group has collaborated with Alfred Hospital throughout the whole process. The Alfred Hospital is one of the advanced tertiary teaching hospitals and is under Alfred Health's management. There are many services provided by Alfred, such as psychiatry, cancer, and asthma. Moreover, Alfred is equipped with a high amount of advanced medical facilities and has a database of GTCS and PNES video recordings. [3]

Founded in 2018, the Monash Medical AI (aka. MMAI) aims to research and applications in medical services based on artificial intelligence. The MMAI consists of many high-achieving researchers and collaborators from software, machine learning, and computer vision backgrounds, which has produced a high publication volume these years. [4]

1.4 Project Background & Motivations

Both the epileptic seizure and the psychogenic nonepileptic seizures are crucial diseases that can lead to severe consequences for the patients. For years, the symptom of these two different seizures has been hard to distinguish by their outward appearance. The most reliable and traditional way to judge is based on EEGs. For the first point, as data discussed in section 1.1, many patients have been misdiagnosed with these diseases due to the incapability of their local medical services. The price of doing a 24-hour EEG monitoring could be more than $600, whereas the deep learning method can dramatically decrease this level once been developed. Infect, besides the IP, the cost of a device using the deep learning method could be just a standard camera and a tiny processing unit. Secondly, the traditional measurement would require wearable sensors to be applied to the patients. This would inconvenience their daily life, and for some special cases, the wearable devices may be dangerous and interfere when the patients are suffering a seizure attack. Given all these conditions and limitations on traditional technology, developing a robust deep learning framework that can detect and classify seizures in real-time will be imperative and meaningful.

1.5 Role in Project

This project is a collaboration project with Dr. Deval Mehta. My role in the project can be split into three major parts. Which is listed as follows:

1. Dataset extraction
   Due to the frontier of this project, there is only a mere epilepsy dataset available online. And since the dataset would directly involve the patients' privacy, acquiring an existing dataset would require a lot of legal documents. Therefore, the demand for building our dataset was raised, following the privacy protection protocol of the Alfred Hospital. The group was only allowed to process video

data supervised by Alfred. My role in this part was to set up and build the pipeline of data extraction on the Nvidia Jetson device. The pipeline includes the whole extraction and storage design of light-weight-human-pose-estimation, dlib-retina face, optical flow, and MediaPipe.

2. End-to-end training process build and validate
   After the dataset extraction and storage process had been built, my role in the next step was to select and modify several repositories that could best suit the project. In this stage, there are more than 5 different repositories have been investigated, and 2 of them have been shortlisted. Their performance was also validated and reported to the research team.
3. Pilot test on the initial dataset
   Finally, in the pilot test part, my role was to test the real performance of all these shortlisted methods based on the first batch of the extracted data from Alfred. And eventually, give feedback on different issues and potential improvement to the research team.

2. Literature Review

2.1 Availability

Until now, quite a lot of research has been done on epileptic detection and classification between epileptic-caused seizures and non-epileptic seizures. Peers has proved that the potential of developing such a deep-learning-based monitoring system is meaningful and necessary. Meanwhile, the voice of the society also has this demand. Except for EEGs, the traditional way of a doctor/nurse to identify seizures was usually based on facial expression; for example, chewing and blinking are two common well-observed patterns of a seizure attack. Body and joint movement are also important decision factors for the professional medical staff. When patients have a seizure attack, their joint movement will become abnormal and can be identified. Head-turning can also be used as a sign. To recognise all these clues and patterns correctly and quickly would require a long duration of medical training and based on a lot of experience. This is terrible news for countries with poor health care and patients who cannot afford the treatment in a hospital. However, based on the study and prior work of [5], it has been shown that automated epilepsy analysis could be done based on computer vision and its relative derivative technology.
From the research done by [6], it is noticed that the traditional way gradually cannot satisfy everyone's demands nowadays. One of the disadvantages of the traditional way is they always rely on a branch of sensors and wearable devices. As discussed in section 1.1, the patient group of epilepsy includes all ages, which means the wearable devices are not safe and tolerable for children and people with intellectual disabilities. In practice, these patients showed repellence and were always trying to dislodge the traditional instruments.

2.2 Deep Learning (DL) and Neural Networks

Deep learning is one of the most important branches of machine learning. It was inspired by the architecture of human brains, which was then implemented and enhanced in programming languages. The most common neural network is multiple layers with many neutrals in each layer concatenated together. To have the ability to process nonlinear tasks, there was nonlinearity added between each of the hidden layers. For example, relu, sigmoid, tanh, leaky relu, etc. As shown in figure 2.2.1 and the corresponding mathematical representations for these four activation functions, they all have their merits and demerits in different aspects: The sigmoid function has smooth transaction and can be derived for all regions. However, it requires a lot of computational effort and is easy to cause gradient vanishing problems. The tanh function is also smooth at transactions and has a mean value of zero; the disadvantage of tanh is the same as the sigmoid function. For relu, the advantage is that it can quickly converge and solve the gradient vanishing problem for the previous two

functions, and the computational effort for the relu function is very simple; the disadvantage of the relu function is it may cause the dead-relu issue (it happened when a huge gradient passed the relu function and then after the backward pass the value in the neural will always lower than zero and not updating anymore). The leaky relu function inherited the advantages from the relu function and solved the dead-relu issue. The next step is to perform the forward pass and backward pass after all these architectures have been built. The parameter of each neural can be updated hereby. The deep-learning method is more like a black box process, where the builder does not know what exactly happens inside the box but to 'let it perform in its way'. It has been well-recognized that the deep learning method now has gradually become a big trend and has its strength in many newly emerged problems. The deep learning method has also shown the world that it has much better performance than many traditional methods in different fields. [7]

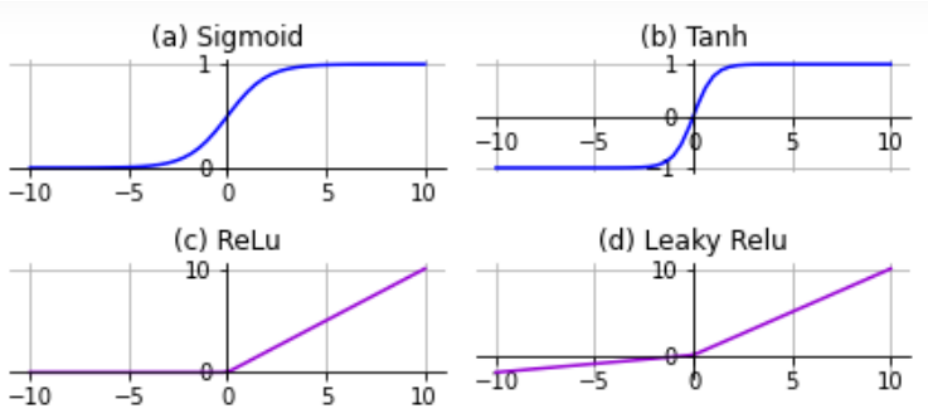


Figure [2.2-1] Most Commonly Used Activation Functions

- Mathematical representations:

$$Sigmoid = \frac{1}{1 + e^{-x}}$$

$$Tanh = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$

$$Relu = \max(0, x)$$

$$LeakyRelu = \max(ax, x)$$

2.3 Semiology of Limbs and Joints

In recent studies, the performance of joint/pose estimation and human movement traction have achieved a robust level. The traction and localisation accuracy can be reliable in a clean and simple environment. Available on GitHub and other platforms, the pifpaf, light-weight-human-pose-estimation, and MediaPipe are the most popular relevant open-sourced repositories to track human joint movement. From the experiment result based on [8], the method they used to decide whether a joint is correctly detected is to calculate the pixel distance between the predicted joint location and the true location. And if the distance is within 8, then this joint data is valid. Even under this strict limitation, the accuracy can still be around 91 per cent. The most challenging part reported by [8] is the wrists and the elbows.

2.4 Residual Network

The idea of residual learning was first brought up by Microsoft Research [14]; the motivation for developing the residual network is to find the solution to the degradation problem of traditional deep learning networks. The degradation problem is a phenomenon in which the performance of deep learning networks decreases as the depth increases, which means simply adding more layers to a deep learning model does not guarantee improvement in terms of accuracy. The degradation problem is different from the gradients vanishing/exploding problem; it has low accuracy on both the training and testing dataset.

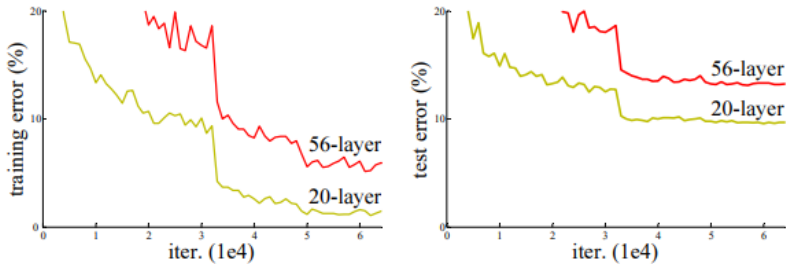


Figure [2.4-1] The training and testing error on the CIFAR-10 dataset [14]

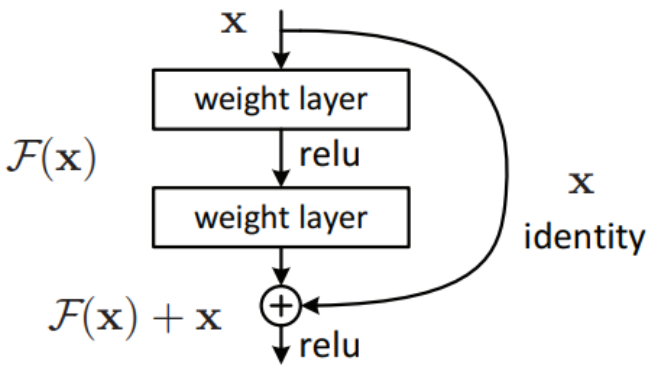


Figure [2.4-2] A segment of the Residual Network [14]

As shown in Fig.2.4-1, as the depth of the model increases, the training, and testing error both get increase. This is the outer look of the degradation problem.

Different from traditional deep learning networks, the residual network was trying to fit the residual error from previous layers instead of only fitting a distribution. Based on the result from [14], the residual network can be a solution to the degradation problem as it can achieve better accuracy as the depth of the network increases. In 2015, the residual network wined the ImageNet Large Scale Visual Recognition Challenge, and it has been widely used in many kinds of datasets with solid performance.

The basic theory of residual network is as follows, define $H(x)$ to be the underlying mapping wanted (it does not need to be the whole network, it can be a mapping from any part of the network). The residual network does not directly fit the model, instead, it is trying to fit the mapping of $F(x) = H(x) - x$. And followed by adding x, we can get the $H(x)$ at the end of the block. The mathematical representation can be achieved by using the 'short cut' shown in Fig.2.4-2, where one can observe that the identity of x is directly passed to the end of the block through a 'short cut'.

Table [2.4-1] Common Structure of Residual Networks [14]

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

2.5 Long-Short-Term Memory Networks (LSTM)

The long-short-term memory networks have become very popular and have gained focus since the 20th century. Based on the data provided by Google, there were more than 16000 citations in a single year. The LSTM is one type of neural network; it inherits the advantages of RNN and fixes the vanishing gradient problem in RNN. Unlike traditional models, the feedback connection is one advantage/variety of the LSTM. This specialty has made the LSTM capable of identifying and learning from a data sequence. For instance, a time series of videos or audio instead of a single image/data sample. The reflection of the publications has shown that the LSTM networks are widely used in speech recognition, video recognition, classification, etc.

Inside an LSTM network, there are three stages. In the first stage, the model 'forgets' some unimportant information that passed by the previous node. In simple words, at

this stage, the model is trying to keep important information in the head and dismiss unwanted/unimportant information. The second stage of an LSTM network is to pick newly emerged important information and then memorise them. And memorise other information based on the weight of how important the information is. At the end of the two stages, the model can add the results together and pass them to the next state. In the 3rd stage of the LSTM, the model would decide which parameter is the output of the current state. Based on a study done by [9], it has been shown the long short-term memory network can fit in the application of epileptic seizure detection. They used the spatial and temporal information extracted from a large number of patients' facial expressions and joint movements. And their results have shown that facial and joint semiology can be properly detected and utilised in LSTM networks to identify different types of seizures. The average test accuracy of their method can reach 92.5 per cent. They also believe their result can be further used as a solid method to assist the clinical decisions.

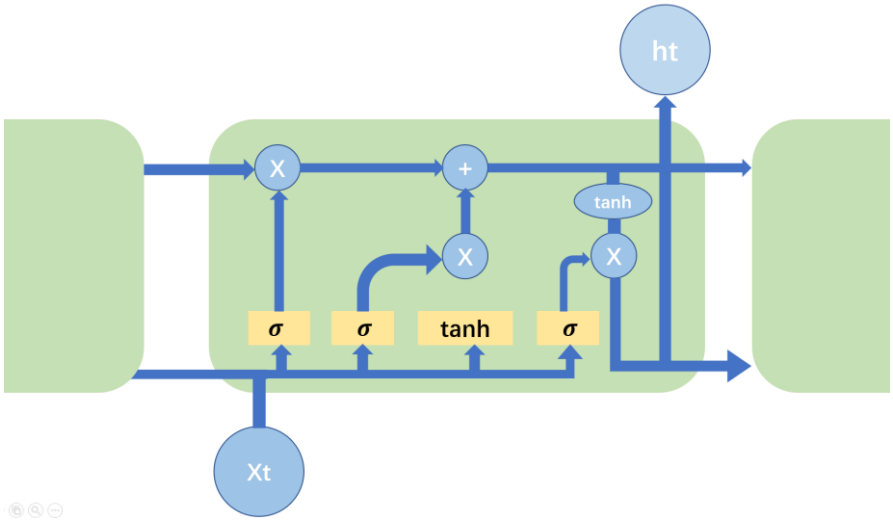


Figure [2.5-1] Structure of a state in LSTM

3. Overview

3.1 Hardware Specification Overview

In the first stage of the project, all the works need to be done on the local PC and remotely on the Nvidia Jetson devices due to the border's restriction. Since topics related to deep learning will usually require high demand on computation power and storage, the local environment has been built with a high standard. The specifications are as follows.

Table [3.1.1] Hardware specification for local PC

Items	**Technical Specification**
GPU	Nvidia GeForce RTX 2070
CPU	Intel Core i7-9750H
Memory	32GB

As mentioned in section 1.5, since the video clips of the patients have high privacy and cannot be downloaded, all the data extraction work is required to be finished at Alfred. The Nvidia Jetson device, therefore, becomes suitable for this project. In this project, we use the Nvidia Jetson Xavier as the platform. It has AI inferencing capabilities on edge devices. The Jetson device has an excellent performance in handling tasks like visual odometry, mapping, object detection, etc. Based on official data, the GPU performance is up to 32 TOPS of peak compute and 750 Gbps of high-speed I/O in a compact form factor. Besides the performance, another merit of the Jetson device is the flexibility. Not only does it have a small size, but also the power profiles can be customised to a specific application. In this project, 2 Nvidia Jetson devices have been set up separately and used in the Alfred. The technical specifications are as follows.

Table [3.1.2] Hardware specification for Nvidia Jetson [10]

Items	**Technical Specification**
GPU	512-core NVIDIA Volta™ GPU with 64 Tensor cores
CPU	8-core ARM® v8.2 64-bit CPU, 8MB L2 + 4MB L3
Deep Learning Accelerator	2x NVDLA
Memory	32GB 256-bit LPDDR4x | 137GB/s
Storage	32GB eMMC 5.1
Size	105mm x 105mm x 65mm

As the project stepped into the second and last stage, we also used M3 for some model training and dataset storage job. The M3 is the 3rd generation of the MASSIVE. It is a high-performance platform with high computing and storage capability. There are a total number of 5673 cores and more than 1.7 million Cuda cores. The total memory of MASSIVE is about 63 TB. In this project, only a few choices of GPUs are tested. Their technical specifications are as follows.

Table [3.1.3] Hardware specification for nodes used on M3

Item (GPU node)	**Technical Specification**
P100	CPU core per node: 28, GPU cores per card: 3584
V100	CPU core per node: 36, GPU cores per card: 5120
T4-Light compute	CPU core per desktop: 6, GPU cores per card: 2560
K80-Heavy compute	CPU core per desktop: 12, GPU cores per card: 4992

3.2 Project Overview

As mentioned in section 1.5, this project is mainly split into three parts, which are as follows.

- Data extraction and storing phase.
- Validation of optical flow and joint semiology phase.
- Pilot test phase.

All the three phases of the project were supervised and suggested by Dr Deval Mehta and Prof Zongyuan Ge. During the whole project, there was a dual connection between the MMAI group and me. The result of each milestone of the project was updated with Dr Deval in time. And suggestions for the next move were feedback accordingly. The connection between Alfred and me is a single direction the most of time. The Alfred only provides the processed result and annotations of the timestamp to the project. The enquiry and demand to Alfred would need to be passed via the MMAI group.

As shown in figure 3.2-1, the video clips of PNE seizures and GTC seizures provided by Alfred are the input of the data extraction block. The whole data extraction block contains three Nvidia Jetson hardware, which are all placed at Alfred. Then the processed data is directly uploaded to the massive 3 project directory. The data storage and data split job were firstly implemented and tested on the local environment and re-implemented on massive. Some relevant works such as statistical analysis and visualisation of the dataset were also implemented in this stage. The method validation was firstly implemented with a relevant public dataset. For instance, the dataset used to evaluate the joint semiology method is based on the NTU RGB+D dataset. And the optical flow method was firstly evaluated on the UCF101 dataset. Based on their performance. During the evaluation process, several refinements to data augmentation and extraction were made accordingly. Finally, the two methods were also evaluated using the Alfred pilot dataset. Since this project at the MMAI group is only at a starting phase, there is not enough data that has been processed and correctly annotated from Alfred's side at this time. So that the evaluation of the Alfred dataset is pilot research, and the result is meaningful for the future study of the MMAI group.

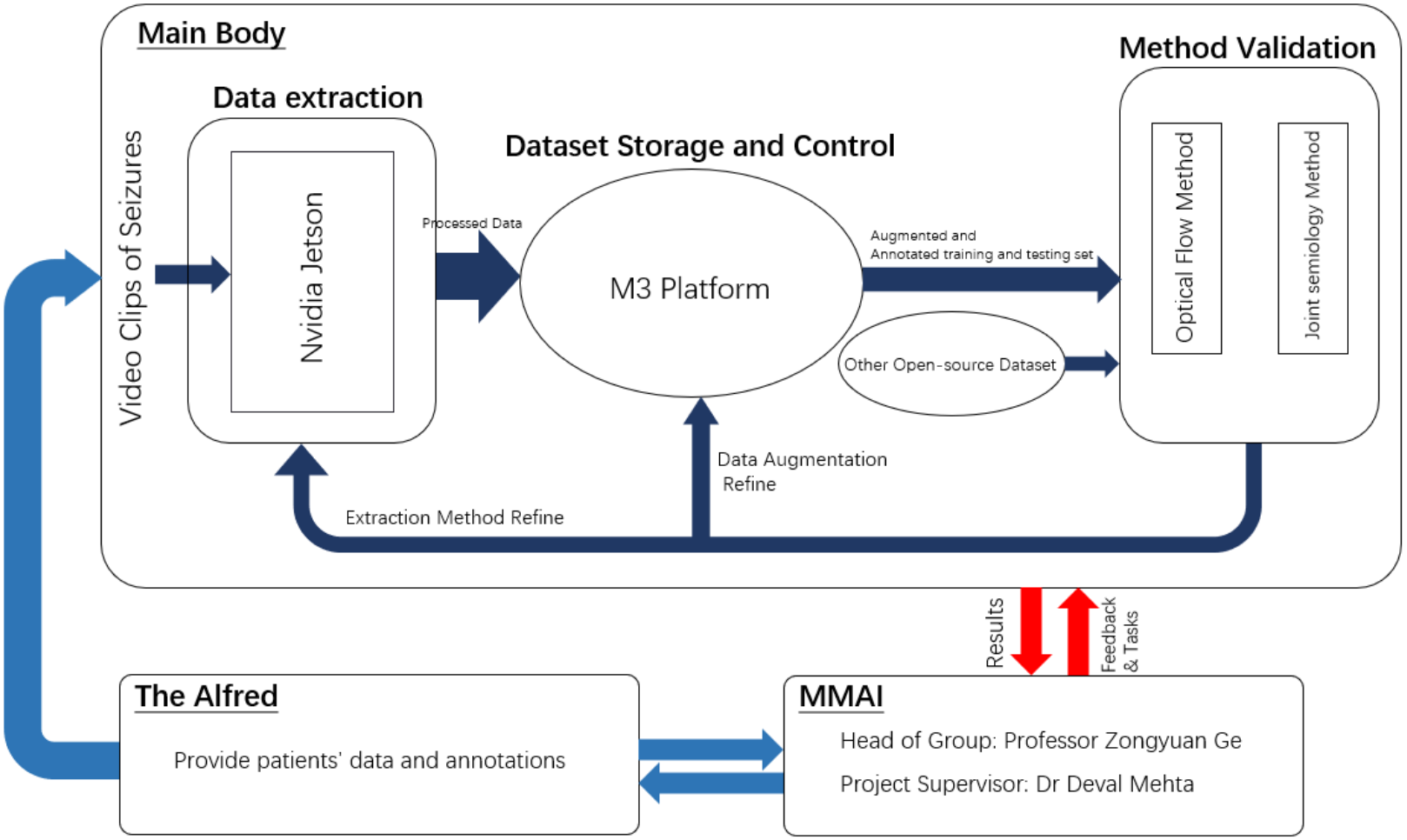


Figure [3.2-1] Project Overview

4. Methodology Specification

4.1 Dataset Extraction and Storing

4.1.1 Environment Setup & Considerations

Due to the border's restriction in the first semester, we have built a remote connection with the Nvidia Jetson boards. Three different remote connection methods have been tested and implemented on the boards: 1. ssh for the command-line interface. 2. Nomachine for visual GUI interface. 3. Use the VNC server to connect the board with a GUI interface.

Among them, the Nomachine method was first implemented and tested. During actual usage and test, the connection with the board is stable, but it was noticed that the connection via Nomachine has a significant lagging issue. This has brought a lot of inconvenience to the development progress. To solve the lagging issue, I have set up the VNC (Virtual Network Console) connection on the board. A VNC usually contains two parts, which are the vncviewer and the vncserver. And it can be used under Linux OS as an open-source. In a VNC connection, the GUI at the board side is firstly compressed and then sent to the control side, the transmission is based on TCP/IP, and the VNC protocol is based on RFB protocol. The theory of the RFB transmission protocol is to run an additional server on the remote devices; the display is transmitted via the RFB and shown on local devices. The VNC session is an own private session. To access the session, RealVNC has been used on the local device. After the VNC connection has been set up, the tested latency has been decreased to 2~4 seconds, which is a huge improvement over the Nomachine connection method. To facilitate some simple tasks, the ssh connection was also made available. The merit of an ssh connection is the speed is very fast. The demerit is it only has a command-line interface.

On the Nvidia Jetson boards, the python environments have been separated to reduce and prevent potential conflict between different repositories. There were initially four isolated environments that were established and managed by pyenv. Since the Nvidia Jetson platform is different from the normal Linux OS, there are a lot of tricky errors have been met during the set-up. For instance, there are a lot of modules and repositories that cannot be directly installed by the pip command.

After researching relevant publications and studies, two methods to perform action recognition and classification were settled from the beginning of the project, which are using optical flow frames and joint-facial semiology as input to the residual networks, respectively. The objective of using optical flow frames and joint-facial semiology is to dismiss unwanted information and pack in more helpful information. Such that it can be easier for the deep learning network to learn the pattern.

4.1.2 Optical Flow Dataset Extraction

Optical flow is commonly defined as an apparent motion of pixels and is a concept used for object motion detection. It was firstly introduced in the 1940s by James J.

Gibson. One of the major applications of optical flow is motion determination and tracking. With the ability to deduce the regions of movement and the velocity of motion, optical flow can handle most of the fields concerning motion analysis, such as analysing relative motion between viewpoint and the observed scene. The optical flow method can measure the instant velocity of each pixel of a moving object on the image plane by utilising the difference between two adjacent frames in a time series to find out the correlation between the two frames. The benefit of the optical flow method is to gain the information of a motion field in a time series of image-dataset, which cannot be done by just analysing frames separately. There are three assumptions for the optical flow method, which are

1. The RGB information of two adjacent frames is similar (for grey images, the brightness of two adjacent frames should be similar).
2. The sample rate should be high, which means the time difference between two adjacent frames should be as small as possible.
3. The motion in the environment should have consistency.

The optical flow method can be split into two based on its usage, one of them is used to fit in the sparse optical flow, and the other is used to fit in dense optical flow. The sparse optical only process a small set of pixels from the whole image and has lower computational efforts. On the other hand, the dense optical flow processes every pixel in the image, it needs higher computational efforts and is expected to produce higher accuracy. As this project is in a pilot study phase and does not need to perform real-time computing, we decided to use dense optical flow to gain more accuracy.

- Mathematical representation:

Define: $I(x, y, t)$ represents the lighting intensity for pixel located at x y at time t. Based on assumption 1, we assume the intensity of this pixel remains the same after dt time unit. Which is:

$$I(x, y, t) = I(x + dx, y + dy, t + dt) \quad (1)$$

Take Taylor's expansion of the R.H.S of (1):

$$I(x + dx, y + dy, t + dt) = I(x, y, t) + \frac{\partial I}{\partial x}dx + \frac{\partial I}{\partial y}dy + \frac{\partial I}{\partial t}dt + \varepsilon \quad (2)$$

Substitute (2) into (1) and divide by dt gives:

$$\frac{\partial I}{\partial x}dx + \frac{\partial I}{\partial y}dy + \frac{\partial I}{\partial t}dt = 0 \quad (3)$$

Let $u = \frac{dx}{dt}, v = \frac{dy}{dt}$ to be vectors that denote the optical flow on the X and Y axis.

Let $I_x = \frac{\partial I}{\partial x}, I_y = \frac{\partial I}{\partial y}, I_t = \frac{\partial I}{\partial t}$. Thus, equation (3) can be rewritten as:

$$I_x u + I_y v + I_t = 0$$

Where u and v are the optical flow represented in vector form.

In figure 4.1.2-1 ~ 4.1.2-4, a moving arrow has been used to demonstrate the graphical appearance of the optical flow. In the original input video, the background has been set to black all the time to make the observation clear. As shown, when the arrow is moving right, the edge of the arrow will be brighter (intensity increasing) in u channel; the edge of the arrow becomes dark in u channel (intensity decreasing) when the arrow is moving left. A similar pattern also exists in v channel: when the arrow is moving up, the edge will become dark, and the edge will become brighter as the arrow moves down. The RGB channel is based on similar logic with colour mapping.

In figure 4.1.2-1, the rows are original image, RGB channel optical flow, u channel optical flow, and v channel optical flow, respectively.

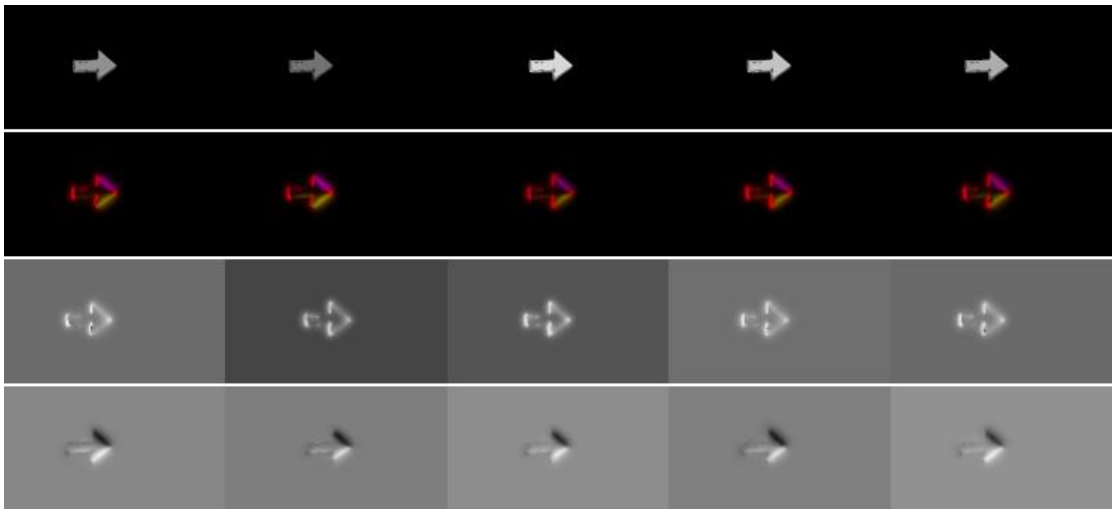


Figure [4.1.2-1] Optical Flow for Object Moving Right

In figure 4.1.2-2, the rows are original image, RGB channel optical flow, u channel optical flow, and v channel optical flow, respectively.

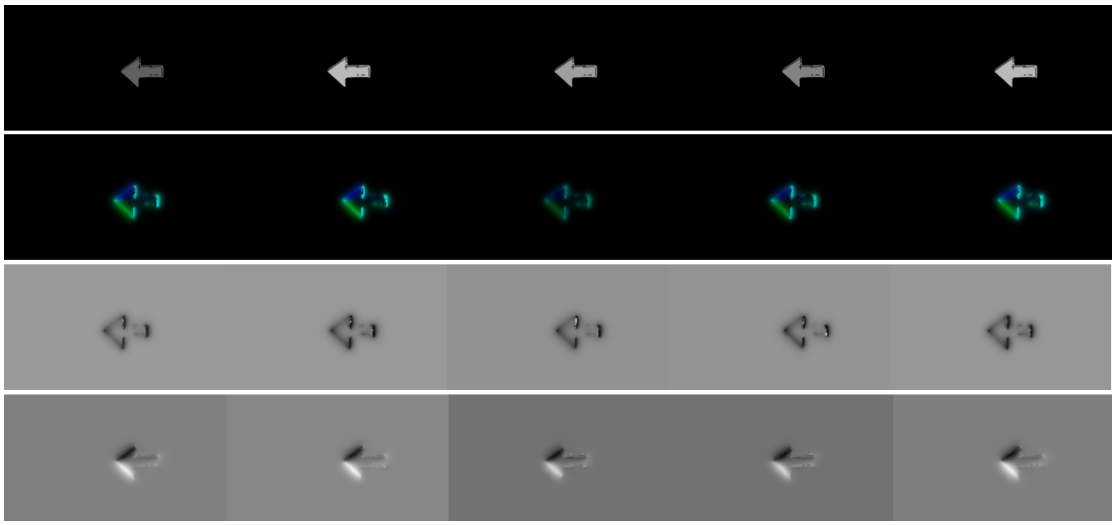


Figure [4.1.2-2] Optical Flow for Object Moving Left

In figure 4.1.2-3, the rows are original image, RGB channel optical flow, u channel optical flow, and v channel optical flow, respectively.

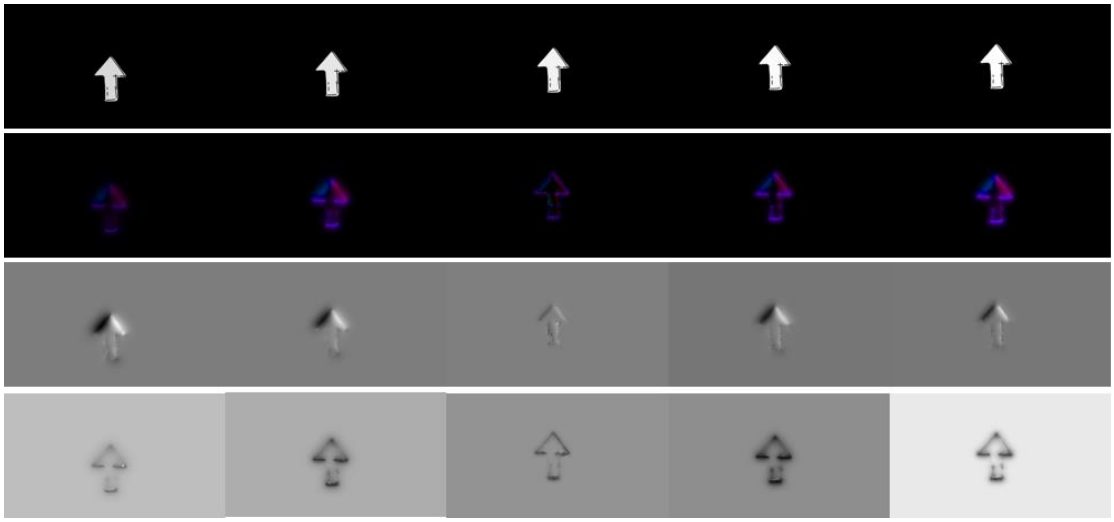


Figure [4.1.2-3] Optical Flow for Object Moving Up

In figure 4.1.2-4, the rows are original image, RGB channel optical flow, u channel optical flow, and v channel optical flow, respectively.

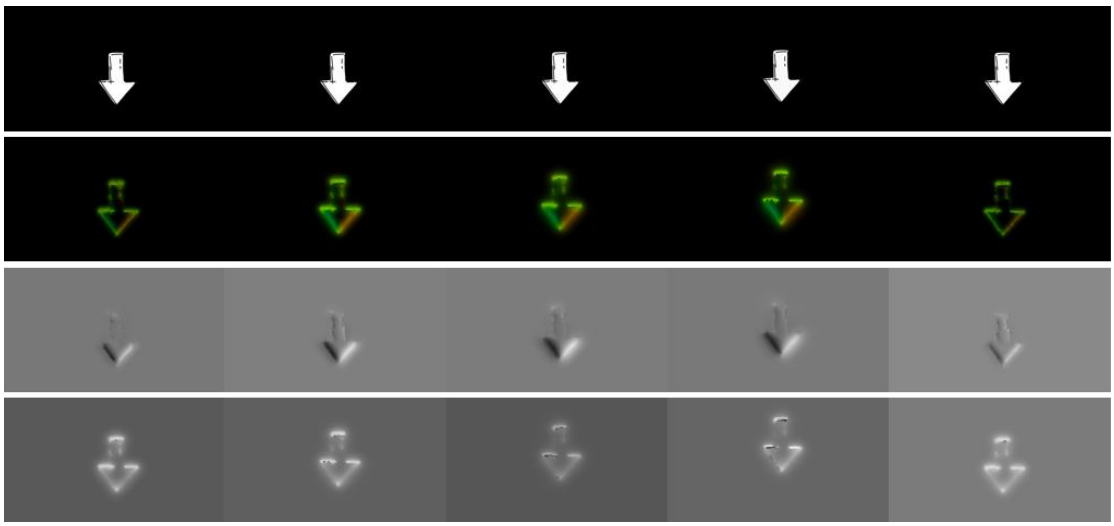


Figure [4.1.2-4] Optical Flow for Object Moving Down

The optical flow extraction block was implemented based on OpenCV. It takes single-channel images for the previous frame and the next frame as input. And outputs the calculated flow image, which has the same shape as the input images. In this implementation, the region of interest is not used during the extraction phase; all the

input frames are the original size of the videos.
In the first version of the optical flow extraction method, only the RGB channel was saved to the M3. However, the deep learning model we used in the second stage of the project would also require u and v channels of the optical images. Since the RGB channel was calculated through a normalisation block, and the normalisation method is cv.NORM_MINMAX, therefore, we cannot revert the calculation. To handle this, we have had to rerun the extraction process in Alfred. The normalisation equation is as follows.

$$v_{normalised} = \frac{v - \min(A)}{\max(A) - \min(A)}\big(newmax(A) - newmin(A)\big) + newmin(A)$$

where $v_{normalised}$ is the normalised value, v is the original value, A is the sample set.

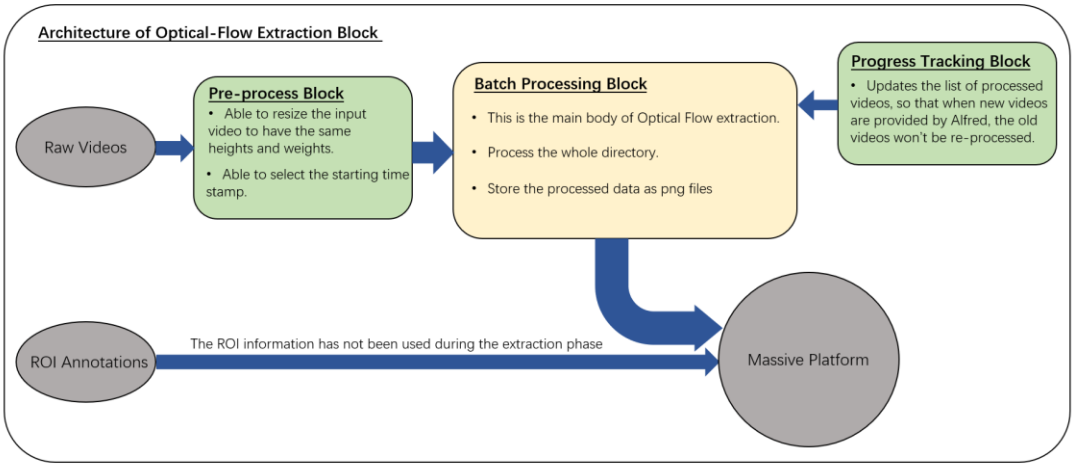


Figure [4.1.2-5] Architecture of Lwpose Extraction Block

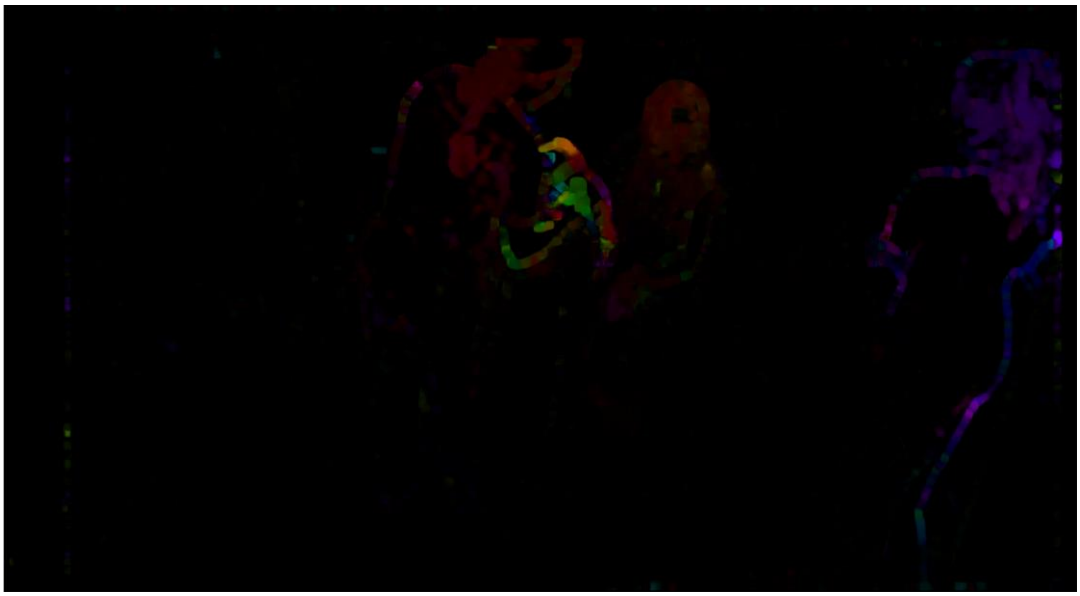


Figure [4.1.2-6] RGB channel of optical flow image

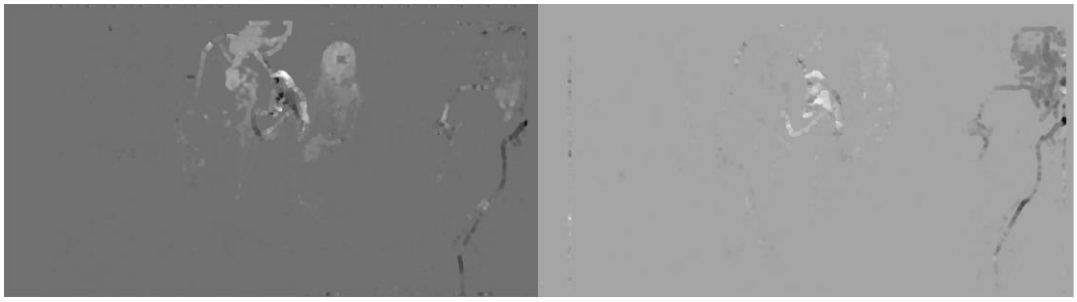


Figure [4.1.2-7] u and v channel of optical flow image

As shown in Fig. 4.1.2-2 and Fig. 4.1.2-3, u and v channels denote the optical flow on the X and Y axis, respectively; the RGB channel is the result of compact u, v channels and then mapping to colour.

4.1.3 Joint-semiology Dataset Extraction

Another data extraction block is built to generate data needed for the joint-semiology method. The most related extraction technic to this topic is linked to human pose estimation. Human pose estimation is a general technic that is commonly related to Computer Vision and Deep Learning. The output of the pose estimation block is a skeleton in a 2D format that represents the orientation of a human. It is also worth mentioning that the 3D format of the skeleton data was also produced and saved for later studies. Once the individual joints are extracted from a picture/video, the image coordinates will be saved along with the joint type. And depending on the joint type, one can decide whether a 'valid connection' has been made. The linkage between two validly connected joints is known as a limb. Pose estimation shows its strength in myriad fields since its invention. Some representative applications are activity recognition and motion tracking/capture. Nowadays, human pose estimation has already become a well-developed and highly reliable field. There exists a wide range of applications related, and most of them can be found as open source. It gives an extra hand to help us go deeper into many fields concerning human activity.

The first repository set upped for joint-head pose estimation is the OpenPose. OpenPose is capable of detecting multi-human skeleton movement in real-time. It was one of the early choices for the project. The performance of the OpenPose has been proven to be very solid. However, this method becomes a bad option from the perspective of computational effort. The computational ability of Nvidia Jetson devices is much less than a normal PC, as the Nvidia devices are aimed as a tiny computation module that can be easily deployed at a low cost. The OpenPose only gives 4 frames per second processing performance; this cannot be tolerated. As if the project/application is eventually deployed into the industry, it would need to do the seizure detection and prediction in real-time. 4 FPS means the model will lose a lot of important data input. The second option for joint-head pose estimation emerged is the lightweight-human-pose estimation (lwpose) repository. Lwpose was invented by Osokin and Daniil; it is a light-weighted modification based on OpenPose. The detection accuracy of lwpose is negligible lower than the original OpenPose model, whereas the computational effort has been reduced. [11] The test result shows that this

method has 7 FPS performance on the Nvidia devices without ONNX. The ONNX is known as Open Neural Network Exchange. This idea was brought out by Facebook and Microsoft. The ONNX is capable of translating any kind of deep learning framework (TensorFlow, PyTorch, One Flow, Paddle). Evidence shows that the lightweight-human-pose estimation can work faster if run under TensorFlow. The translation process in this project is PyTorch → ONNX → TensorFlow. Based on my observation of our Nvidia devices, the FPS has increased to 17, which is more than 4 times better than the beginning. One drawback of this method is the limitation on the input size of images. To run on the TensorRT interface, we need to unify all of the videos to have the same height and width. Finally, the size has been chosen as 640x360. This result is acquired based on different trials. It is noticed that this resolution won't lose too much information while it can be processed with a reasonably good speed. A data visualisation method was also implemented to use as a validation tool after we received all the human joint semiology data on the M3. To facilitate the processing, I also implemented ways to let us directly jump to the start of useful video sections. Based on the first trial experiment done on 17/9/ 2021, we have noticed that in the real clinical environment, some small objects that are close to the patient can be detected as part of the body, and the surrounded medical staff could also mislead the detection. To fix this issue, we have added a Region of Interest selection mechanism (each ROI file contains 4 elements: the x and y coordinates of the top left corner of the bounding box and its width and height). By applying the ROI, we can further specify an area on the image plane that is valid to be detected by the extraction algorithm. Thereby preventing the disruption from the surrounding instruments/medical staff. The following figure shows the architecture of the whole lwpose extraction block.

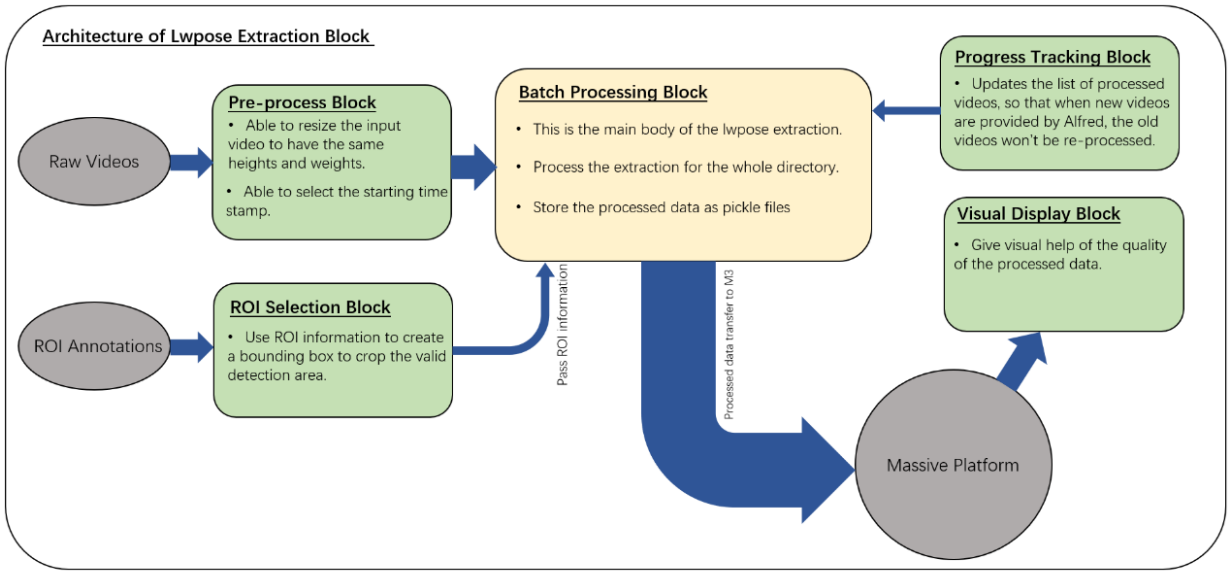


Figure [4.1.3-1] Architecture of Lwpose Extraction Block

The Google MediaPipe is implemented to give the semiology on hands and facial key points. It gives extra information than the Lwpose. The body semiology of MediaPipe has also been saved to pickle files for comparison with the lwpose body semiology

extraction. In the end, the lwpose semiology of the body will be integrated with the facial and hands semiology extracted by MediaPipe. All the body, facial, and hand semiology for each frame will be transformed and saved in relevant pickle files.

4.1.4 Dataset Split

Each video provided by Alfred has three segments: 30 seconds pre-seizures, seizures, and 30 seconds post-seizures. The timestamp for the transition has been given. And the fps for all the videos is 30. All the pre-seizures and post-seizures segments were treated as non-seizures data in this project. And the segment when seizures happen was labelled as GTC seizures and PNE seizures accordingly.

One important consideration is that the actual video length slightly varies with the time stamp information provided by Alfred. If the actual length of the video is shorter than the expected value, and we continue to use the 30 seconds threshold to fetch pre and post-seizure frames, an index out-of-bounds error will emerge. Therefore, the final design collected only the pre-29 seconds and post-29 seconds of video segments as the non-seizures' dataset.

There is a second dataset split methodology, which further splits the labelled segments into smaller pieces. For example, after performing the first-stage dataset split on one of the videos, we will have two non-seizures video segments and one seizure video segment. Followed by this, the video segments can be further cut and split by every 10 seconds. The initial intention of using this method is because of dataset insufficiency. Before the further split, the dataset only contains 9 seizure samples and 18 non-seizure samples. By applying the further split, the dataset has been increased to have 105 samples in total. One disadvantage that cannot be neglected is doing such a further split may lose the internal pattern of how seizures gradually break out. Therefore, this further split method can only be tested during the pilot research.

Number	File ID	Video FileName	Event Date	Seizure Classification (Detailed)	Seizure Onset	Side	Export Start Time	Export End Time	Video Duration	Event Start Time	Event End Time	Event Duration	EEG Onset Time	Clinical Onset Time	Seizure End Time
BLOCK1															
1	diazA_12-Dec-2(	diazA-12_Dec_	#######	Focal to Bilateral Tonic-Clc			5:18:19	5:23:04	0:04:45	5:18:49	5:22:34	0:03:45	5:18:49	5:19:42	5:22:34
2	hendS_7-Mar-20	HENDS-6_Mar_2019	#######	Focal to Bilateral Tonic-Clc			6:57:05	7:00:14	0:03:09	6:57:35	6:59:44	0:02:09	6:57:35	6:58:08	6:59:44
3	johnD_11-Apr-2(	JOHND-10_Apr-2019	#######	Focal to Bilateral Tonic-Clc			6:32:48	6:38:47	0:05:39	6:33:18	6:38:17	0:04:59	6:33:18		
4	knigT_12-Jul-2(	KNIGHT-11-Apr_2018	#######	Focal to Bilateral Tonic-Clc			4:55:13	4:57:13	0:02:00	4:55:43	4:56:43	0:01:00	4:55:13		
5	sedaB_12-Nov-2(	sedaB-12_Nov_2018	#######	Focal Aware Seizure			22:29:00	22:34:00	0:05:00	22:31:23	22:32:19	0:00:56			
6	langP_4-Dec-20	langP_4_Dec_2018	#######	Focal to Bilateral Tonic-Clc			15:52:00	15:55:02	0:03:02	15:52:30	15:54:32	0:02:02	15:52:30	15:52:32	15:54:32
7	crosK_7-Jul-20	CROSK-7_JUL_2021_VF	#######	Focal to Bilateral Tonic-Clc			23:35:40	23:39:15	0:03:35	23:36:10	23:38:45	0:02:35	23:36:10		
8		crosK-7_JUL_2021_VS													
9	densE_7-Apr-20	DENSE_7-Apr-2021_VF	#######	Focal to Bilateral Tonic-Clc			6:17:19	6:19:56	0:02:37	6:17:49	6:19:26	0:01:37	6:17:49	6:18:14	6:19:26
10		DENSE_7-Apr-2021_VS													
11	sainM_19-Feb-2(	SAINM-19-Feb-2020	#######	Focal to Bilateral Tonic-Clc			3:04:15	3:07:43	0:03:28	0:04:46	0:07:13	0:02:27			

Figure [4.1.4-1] Example of Dataset Time Stamp

As Fig.4.1.4-1 shows, the 'Video Filename' column denotes the name of each video sample, the 'Export Start Time' column is the time we start saving output, and the 'Export End Time' column is the end time of collecting output, and the 'Event Start Time' and 'Event End Time' is the head and tails for the seizures. It can be observed that the 'Export Start Time' is 30 seconds before the seizure, and the 'Export End Time' is 30 seconds after the seizure.

4.2 Deep Learning Models Validation

At the pilot testing stage of the project, the collaborator at Alfred had met a problem regarding the incorrect timestamp of annotation. This issue has existed for a long time, which resulted in the pilot test cannot being started. To use the time efficiently, we decided to add a deep learning model validation stage to the original plan. During that time, I used public data set to test the availability of the deep learning models. Two models have been implemented to test the performance of the optical-flow-based and joint-semiology-based action recognition and classification. Which are 'two-stream-action-recognition' [12] and 'HCN-pytorch' [13]. The reason for choosing them is because the dataset format of these two models is close to ours, and they both reflected a good overall performance.

4.2.1 Two-Stream-Action-Recognition

This repository uses the UCF101 dataset to perform the action recognition based on a 'two stream' (which are spatial stream and motion stream) method. The UCF101 dataset contains 13320 videos from YouTube and there are 101 different action types, which can be further classified into 5 groups: human-object interaction, human-human interaction, playing instruments, doing sports, and joint movement. The total length of this dataset is 27 hours. Since the original video frames of the patient are forbidden, the spatial stream cannot be utilised. In this project, only the motion stream has been implemented and tested.

Figure [4.2.1-1] UCF101 Dataset

The deep learning network used in this model is Res101. For motion stream, the input is stacked optical flow images which have 10 x channel and 10 y channel. Therefore, the input shape is 20×244×244. It can be treated as images with 20 channels. The x and y channels are u and v channels discussed in section 4.1.2, which can be generated by the optical flow extraction block. The training strategy is to select a batch of videos randomly and then randomly pick one stacked optical flow data sample in each of them.
In the 'motion_cnn.py' the 'Motion_DataLoader' class will take batch size, num workers, in channel, labels, training and testing list, and dataset path as input. Then the 'UCF101_splitter' will be called, which loads the training and testing list, and then returned with two dictionaries containing the video name and corresponding class index (e.g., 1 for playing tennis, 2 for jumping.). The 'Motion_DataLoader' then loaded the frame number for each video from a pickle file. And the frame number is then joined to the end of each video name to be used later. One important function related to the test method is the 'val_sample19'. It will uniformly sample 19 frames inside each test video and save the index of these 19 frames into a dictionary. This dictionary will be the input argument of the 'motion_dataset'. Inside the 'motion_dataset', a 'stackopf' function is used to stack optical flow images. When it is in validation mode, it will stack 10 consecutive u and v channels of optical flow images starting from every 19 indexes (i.e., each sample will have a shape of 20*height*width). The prediction of one video is based on the voting result of the 19 stacked optical flow images. When in training mode, the 'motion_dataset' will randomly select an index in the training video and generate a stacked optical flow image using the same method. Hence, the input data to this network can be treated as a 20-channel image.

4.2.2 HCN-pytorch for Action Recognition

Another method that has been validated for action recognition and classification is the HCN-pytorch repository [13]. It uses skeleton data to do the co-occurrence feature learning. The reason for choosing this repository is under the consideration of data format. In the first stage of the project, the human joint semiology was obtained and saved on M3. These data can be easily transformed to feed in the HCN model. The dataset we used for validating the action recognition performance is based on NTU-RGB+D. There are 60 types of activities included in this dataset, with 56880 samples in total. The NTU dataset used two different methods to split the training and testing set, 'Cross-subject' and 'Cross-view.' The 'cross-subject' method is based on human ID to split the training and testing set. The training set is from ID of {1, 2, 4, 5, 8, 9, 13, 14, 15, 16, 17, 18, 19, 25, 27, 28, 31, 34, 35, 38}, and the data in remaining ID is used as testing set. The 'cross-view' method uses camera ID to split the dataset. Both methods aim to give the model more robustness and become more convincing. The validation experiment is done on the Massive 3 platform to make the model reusable in the future. The NTU-RGB+D dataset is also uploaded and extracted.

4.3 Pilot Test

In the last four weeks of this project, the first batch of the correctly annotated dataset has been handed to us, the pilot test is aiming to give an initial intuition of how good the performance of the own dataset will be. The result of the pilot test can give a valuable reflection on the future improvement direction of this project.
The pilot test is only done on the optical-flow-based action recognition. Because the quality of the human joint semiology does not meet the expected level and has a lower chance of success. The detailed outcome of human joint semiology and its relevant analysis will be discussed in section 5.1. The 'two-stream-action-recognition' repository is one of the state-of-the-art methods for action recognition using optical flow images; it will be used and modified in the pilot test.
There are 4 different comparisons have been made for optical-flow-based action recognition. The experiments aim to test the impact on the performance by changing a branch of important factors. The comparisons are as follows:

1. The influence of using different stack sizes of the optical flow images.
   Currently, the stack size used for baseline is 10, which means there will be 10 x channels and 10 y channels being stacked and passed as a sample. As discussed in section 4.2.1, since the training strategy is to randomly pick 1 stacked image in every batch of videos, it implies the model will only see how the pose/motion changed only in 10 consecutive frames. Normally, the frame rate of a monitor/camera is 30 per second, 10 frames will happen in 0.3 seconds. The motion types contained in the dataset used for the validation stage are usually simple motions. For instance, typing and skiing. However, the motion of seizures is much more complex and harder to distinguish. Most of the seizures will take time to develop, so the patient's behavior at the start of a seizure attack could be different from the midterm seizures. Considering this, the performance and influence of increasing the stack length are worth a try.
2. The influence of using a different number of validation samples.
   As briefly discussed in section 4.2.1, the validation method is by uniformly selecting 19 frames in a testing video and stacking each of the 19 frames with the following consecutive 10 frames. Because the minimum test video length is 28, therefore, 19 has been chosen to be the number of uniform samplings such that the index will not be out of bounds. In this project, the video length of a seizure is usually about 2 minutes, the FPS is fixed at 30, thus, there will be around 3600 frames. Therefore, the original validation method may be unsuitable here, as the prediction is based on the voting result of the 19 samples in the video. Intuitively, using more testing samples should give a better prediction. The result will be shown in section 5.2.
3. The influence of using different sizes of residual networks.
   A comparison between the different sizes of residual networks will also be made. Currently, there is an insufficient amount of data that can be used, therefore, the residual network with 101 layers may not be suitable here. The resnet101 is a quite large network, which usually requires a high amount of training data. Based

on this situation, we decided to further test the performance on resnet18, resnet34, and resnet101.

4. The influence of doing a second stage dataset split.
   One significant difference between the Alfred dataset and the UCF101 dataset is that the length of the UCF101 dataset is mostly around 200 frames. If the FPS is assumed to be 30, it will be 6 seconds. In contrast, the dataset generated for epilepsy detection and classification is usually about 2 minutes. As section 4.1.4 discussed, the second stage of dataset split is to further split a labeled sample into smaller pieces. The initial consideration is that seizures are a gradually developed symptom. The patient's behavior may have difference between the start and mid of seizures. And these patterns may be the key factor in distinguishing GTCS and PNES. However, as the 'two-stream-action-recognition' method is already broken up the overall grasp of a complete video sample, so by doing the second stage split will bring no more disadvantages. The advantage of doing so is to increase the dataset capacity.

5 Results and Discussion

5.1 Data Extraction Results

5.1.1 Optical Flow

From the results obtained in September 2021, the extracted features for optical flow showed a robust overall quality. The shape of patients during movement can be clearly observed. The channel split has also been succussed, where u and v channels have correctly saved along with the RGB channel.

As shown in Fig 5.1.1-1 and 5.1.1-2, these are the examples of extracted features in u and v channels in a consecutive 5 frames. It can be observed that how the pose of the patients and medical staff is changing with time. The image is Gray scaled. Grayscale images will only have one channel (intensity) to represent the feature. The value in the channel can vary from 0~255.

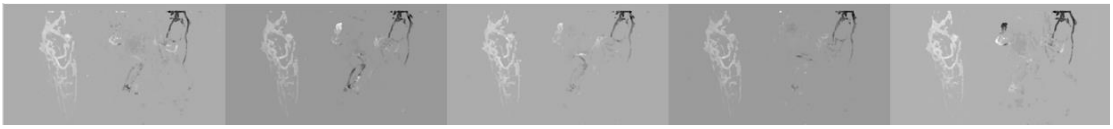


Figure [5.1.1-1] U Channel Optical Flow Features in 5 Consecutive Frames

Figure [5.1.1-2] V Channel Optical Flow Features in 5 Consecutive Frames

Fig 5.1.1-1 and 5.1.1-2 showed the typical situation for the extracted features, in which the medical staff will be captured in the feature plane. This would significantly influence the training and prediction accuracy. In Fig 5.1.1-3, it shows the location of two medical staff and the patient.

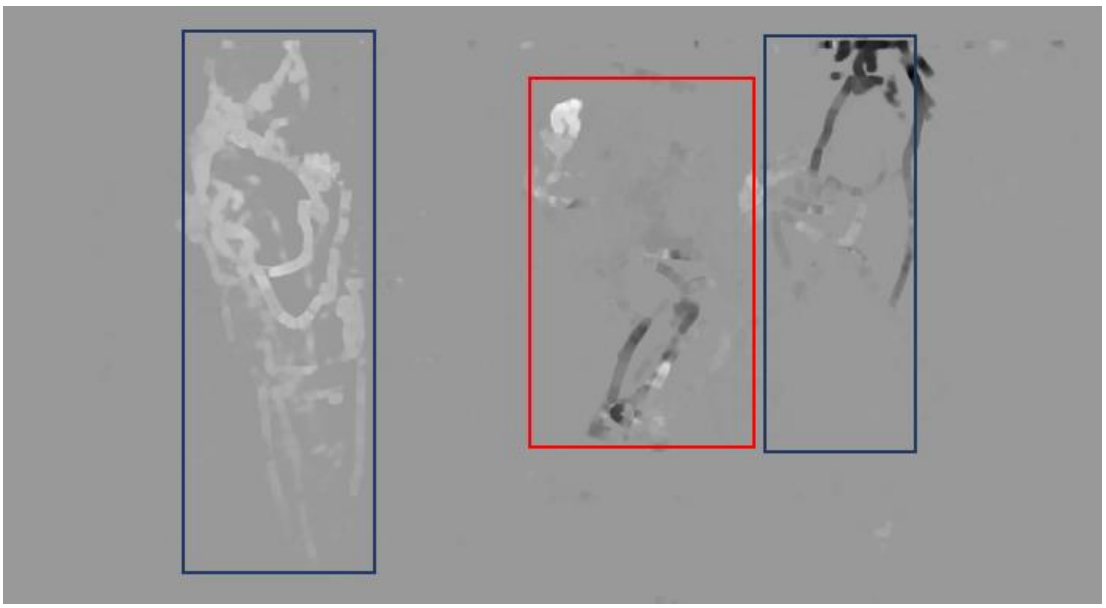


Figure [5.1.1-3] Medical Staff and Patient Location

In order to solve this kind of disruption, the ROI has been added to the dataset creation mechanism. All the videos provided by Alfred were recorded by fixed position cameras, which means the patients' location will be fixed for each video. Therefore, the ROI mechanism can be simply achieved by providing coordinates of patients in each video. The coordinates information was provided by Alfred. After applying the region of interest mechanism, the frames have been cropped only to contain the patient. The output after using ROI is shown as follows (Fig 5.1.1-4 and 5.1.1-5). The features shown in the middle of the image are from the patient. In this way, we have filtered out the interruption in the surrounding environments.

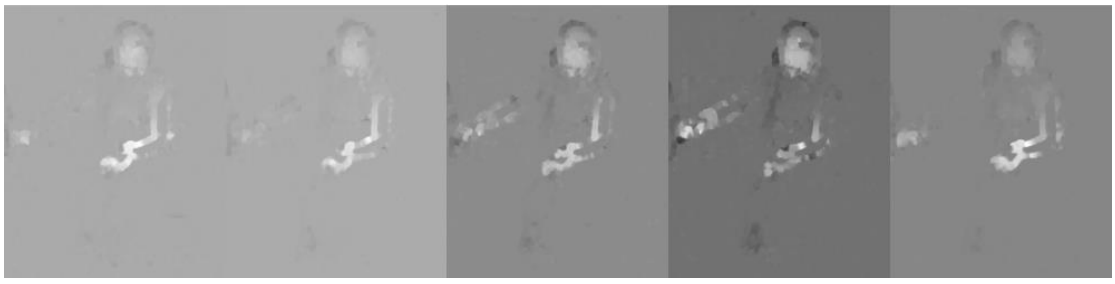


Figure [5.1.1-4] U Channel Optical Flow Features with ROI in 5 Consecutive Frames

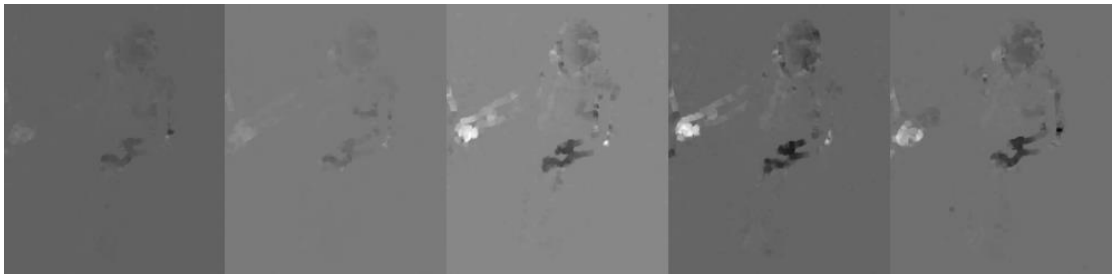


Figure [5.1.1-5] V Channel Optical Flow Features with ROI in 5 Consecutive Frames

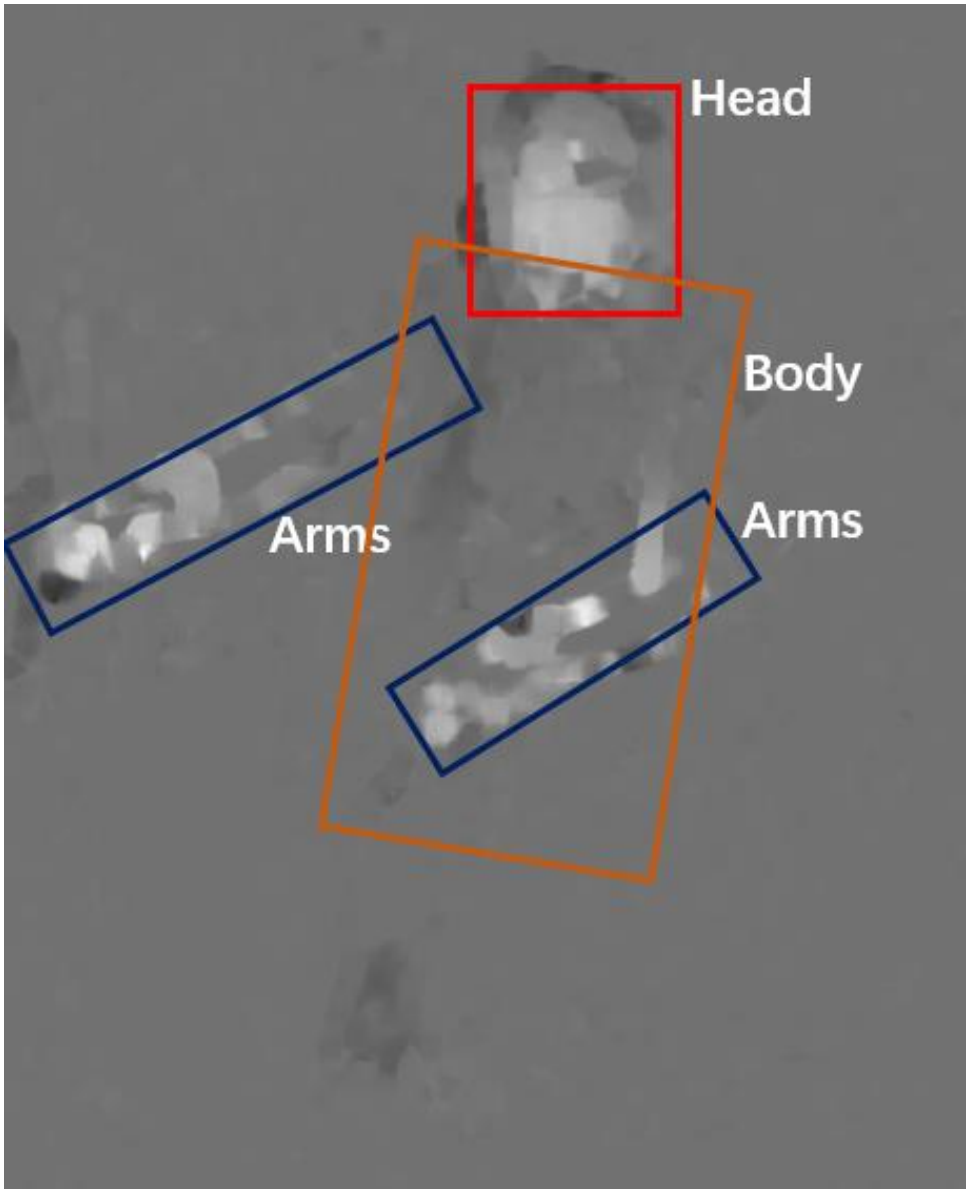


Figure [5.1.1-6] Optical Flow Features

Fig 5.1.1-6 illustrates the detailed body parts on the image, where the red box bounds the location of the patient's head, and the blue box bounds the location of the patient's arms, the orange box shows the rough location of the body of the patient. In this sample, the lower limbs of the patient may be covered by a quilt, so it is not reflected. However, this sample is still quite desired; it included the two limbs and head. The motions on these parts are the key to detecting seizures and making the classification.

The previous examples have shown the desired situation, where the patient's motion can be easily reflected. Unfortunately, in many samples, medical staff would continuously block the scene of the patients. In such sicarios, the joint motion of the patient is no longer visible. And the movement of medical staff will also influence the training and prediction. There is no solution to this kind of defection, the only way to improve this blocking issue is to prevent it from happening. The following figure

5.1.1-7 and 5.1.1-8 shows two examples of medical staff blocking the patients. The red box shows the location of the patient, and the blue circle shows the location of the medical staff. In these cases, the interruption from the medical staff cannot be filtered even the region of interest has been applied.

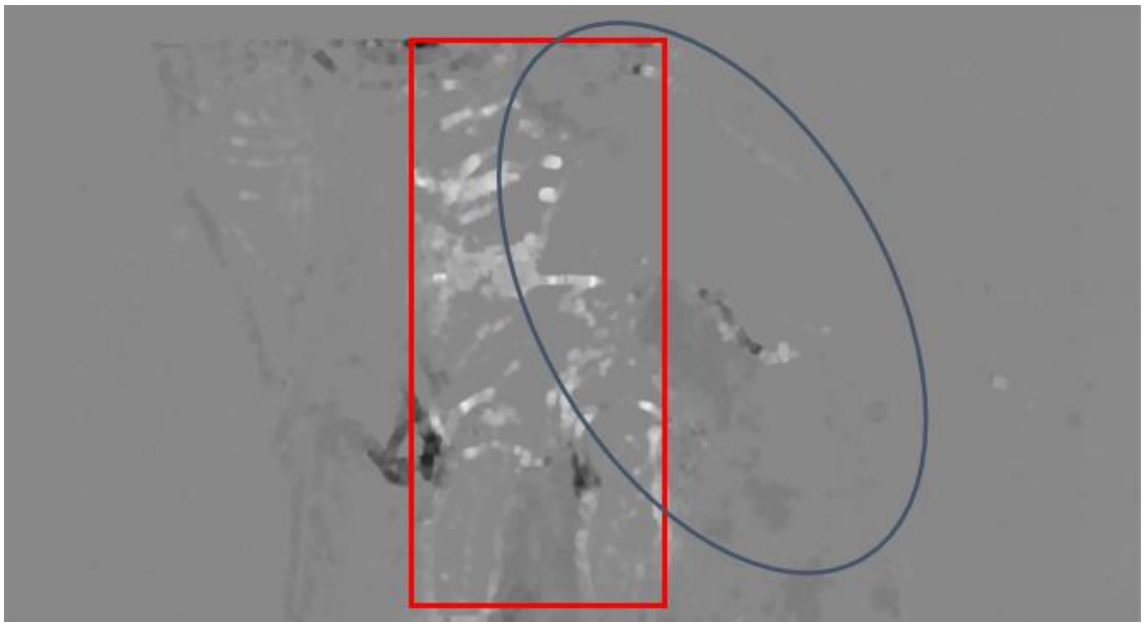


Figure [5.1.1-7] Example of Medical Staff Blocking Patients

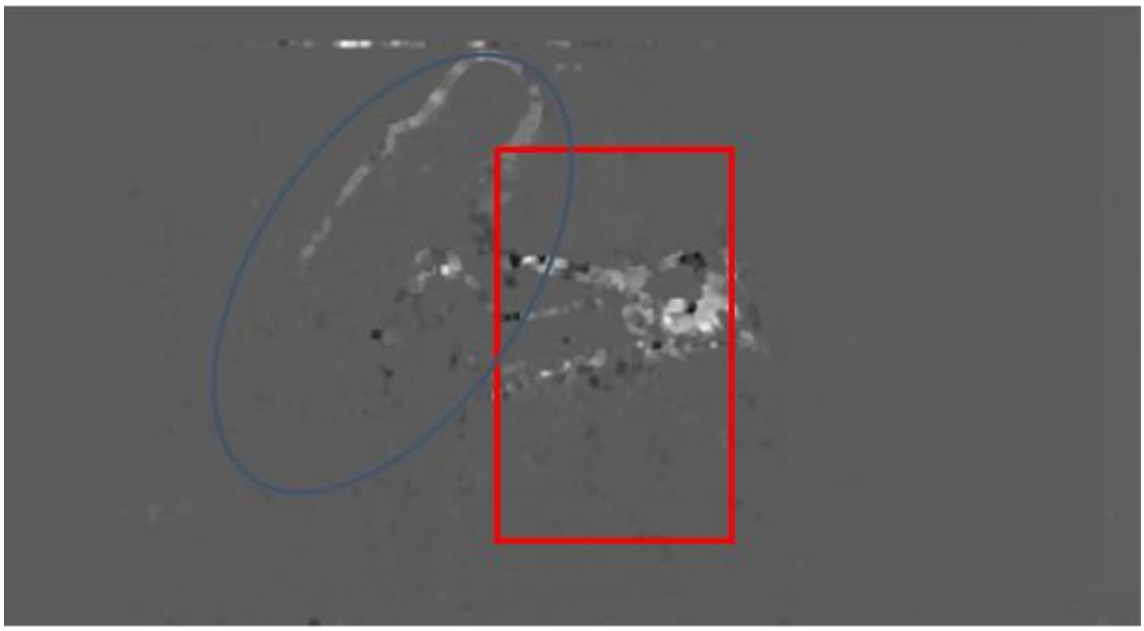


Figure [5.1.1-8] Example of Medical Staff Blocking Patients

Another type of data defect is shown in Figures 5.1.1-9. Alfred has provided some video recorded during the night. The lighting conditions are poor in these samples, which causes the feature extraction to be underperformed. As can be seen in the following figure, the optical flow is quite weak under low illumination conditions. Some videos recorded in Infrared cameras have a similar issue.

Figure [5.1.1-9] Example of Optical Features under Poor Lighting Conditions

The last type of data defect is caused by noise in the original video. The videos used in this project span several years, some of the videos are recorded with a low-resolution camera and may not be stored properly. It is highly likely some videos have used lossy compression, which makes a lot of noise been added. The noise is observed across the whole video (including pre-seizures, seizures, and post-seizures parts) and looks like white noise. Such phenomena could impact the training and prediction accuracy, as the shape of human motion becomes fuzzy. Fig 5.1.1-10 illustrates the phenomena mentioned.

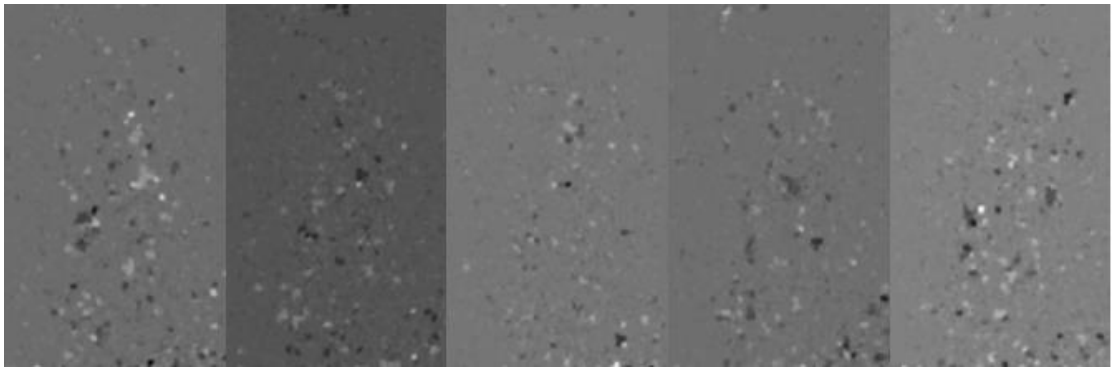


Figure [5.1.1-10] Example of Optical Features with Noise

In general, the data extracted for optical flow has a reasonable quality. Especially in some video sample we have obtained quite desired results, which clearly showed the

important features for detecting seizures. There are some data, however, have deflects cannot be removed. These data are valuable for the next stage of development of this project. Video with low resolution and poor lighting conditions should be discarded as much as it can be. More than that, when recording the video, the camera position needs to be carefully considered to minimise the blocking issue, if safety can be granted, the medical staff should maintain distance to the patient. So that no other motion will be recorded to mislead the judgement of the deep learning model.

5.1.2 Joint-semiology

Based on statistical results (shown in the following analysis), the ‘light-weight-human-pose-estimation’ was used to extract all the body joints, and the ‘MediaPipe’ was used to extract all the facial and hand joints. In the final version of joint-semiology extraction, the region of interest mechanism has been applied.

The first batch of joint-semiology extraction results is obtained at the same time as optical flow. The following figure shows one of the examples of desired results, where all the body joints have been recognised with reasonable error. And the MediaPipe has correctly recognised the face and the hand of the patient. Note that the model can recognise multiple hands in one frame; in this example, only one hand has been recognised. Fig 5.1.2-1 shows the desired results of joint-semiology extraction. Fig 5.1.2-2 shows how the two detection algorithms integrated.

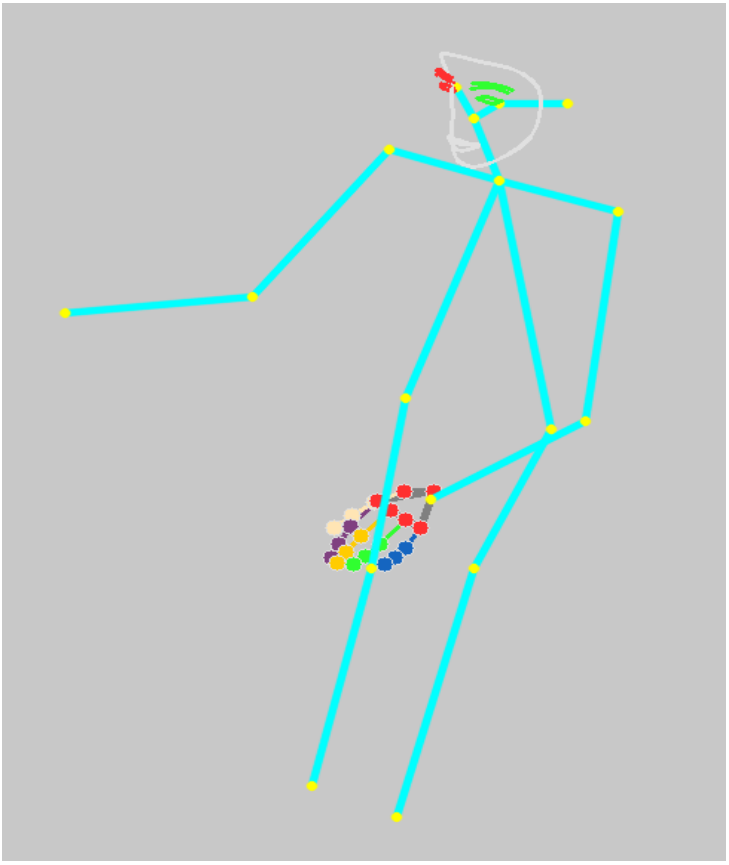


Figure [5.1.2-1] Desired Features of Joint-semiology

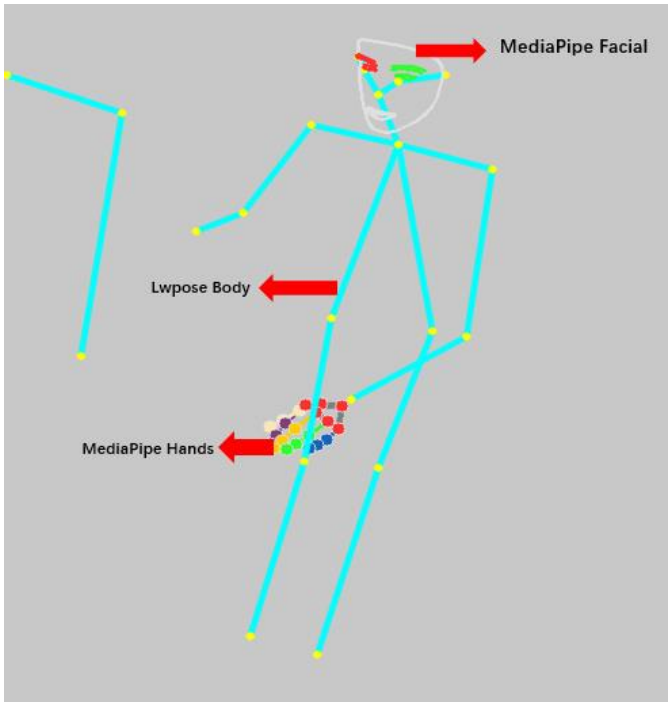
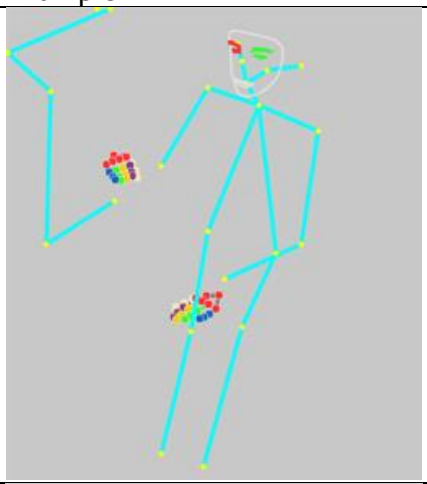
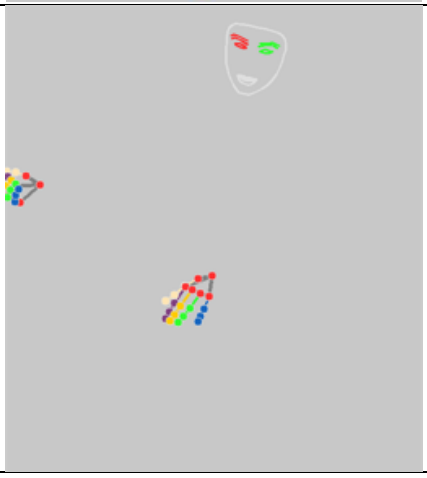
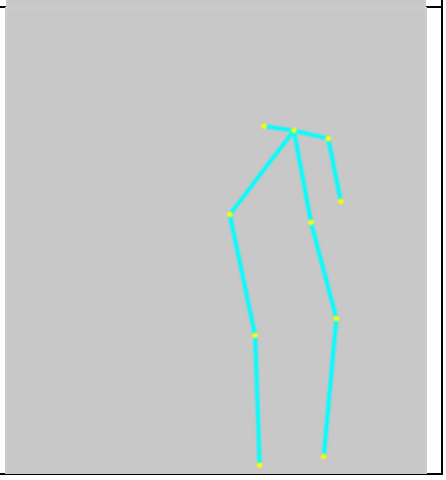


Figure [5.1.2-2] Integration of MediaPipe and Lwpose

The most common deflects seen in joint-semiology features are listed as follows.

Types	Example 1	Example 2
Multiple Persons Detected		
Lost Key Points		

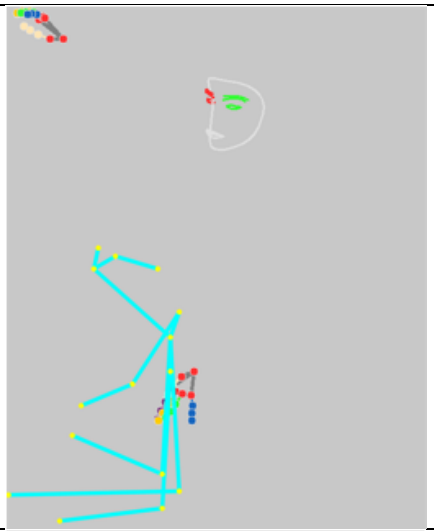
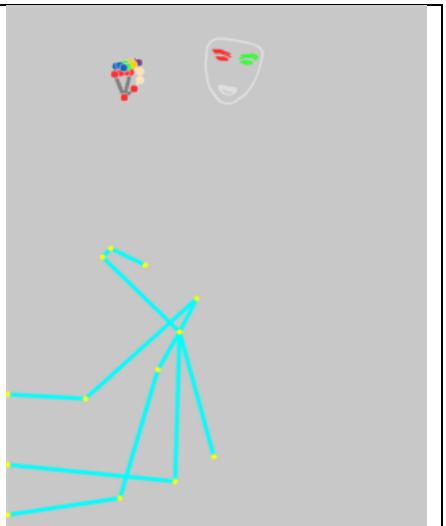
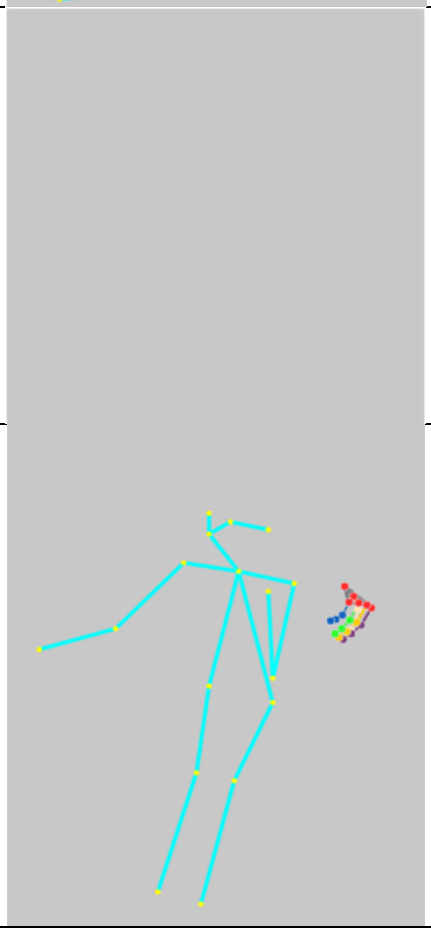
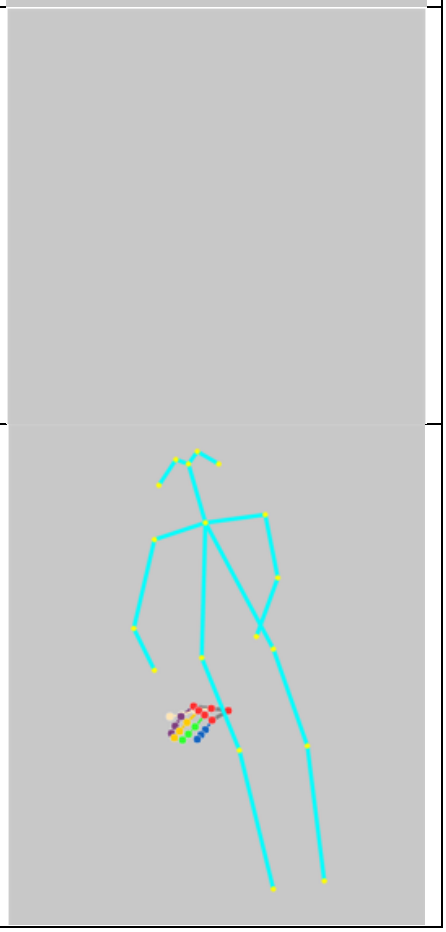
Wrong Key Points		
No Detection		
Conflicts		

Table [5.1.2-1] Most Common Types of Error

Table 5.1.2-1 illustrates the 5 most common types of deflect (undesired pattern) that can be observed from the joint-semiology dataset.

1. Multiple persons detected
   This type of sample is due to the similar issue mentioned in optical flow results. The patients under seizure cannot control their body movements. Therefore, they may be entangled with some wearable sensors and hit medical instruments. From a safety perspective, medical staff will clean the surrounding environment of the patient, and they will be recorded even with the help of ROI.

2. Lost Key Points
   This type of deflects has more occurrence than type 1, where some joints of the patient are not detected/extracted. It is mainly due to the low resolution of the input video and the long distance between the camera and the patients. There will be a separate table to illustrate this issue.
3. Wrong Key Points
   This type of deflects is fatal to the model, as the algorithm wrongly considered some other objects as human joints. (e.g., bottles happen to be misrecognised to be part of human bodies and cause the pose to become discontinuous.) The deep learning model used to detect seizures will be disturbed if a large number of wrong key points have been fed in.
4. No Detection
   In some extreme conditions, for instance, during the night, the algorithm to detect body joints, head joints, and hands cannot detect any skeleton from the patients. However, this kind of deflect can be found by checking the joint number in each frame. Then those data samples can be discarded.
5. Conflicts
   The last type of data deflect is conflicts. This is the case when results obtained from 'light-weight-human-pose-estimation' mismatches the results obtained from 'MediaPipe'. And it is tricky to decide which algorithm has a higher probability of being correct. As shown in the figure, the position of the hands is far away from the wrist position.

The statistical results for part of the processed video are shown in the following table. It shows the missed joints from each algorithm for each input video. The result shows the face and hands joints are the hardest to extract; more than half of these joints have not been extracted from the video. In terms of body joints, Lwpose has better performance than MediaPipe. The missing rate for Lwpose body, MediaPipe face, MediaPipe body, and MediaPipe hands are 15.7%, 67.4%, 23.9%, 63.5%, respectively.

Table [5.1.2-2] Statistic Analysis for Missed Joints

Video ID	Lwpose body missed	MediaPipe face missed	MediaPipe body missed	MediaPipe hands missed	Total Frames
DENSB-7_Apr_2021_VS	791	4729	732	4670	4929
sedmB-12_Nov_2018	0	3727	6	290	9651
JOHND-10_Apr_2019	2479	8161	4627	8880	11024
HENDS-6_Mar_2019	121	764	206	4127	6015
DENSB-7_Apr_2021_VF	500	4659	1702	4480	4955
diazA-12_Dec_2019	2197	8640	2817	5595	8828
KNIGT-11_Apr_2018	1699	2591	1745	3322	3933
Overall	7787	33271	11835	31364	49335

Another issue we noticed in the joint-semiology extraction result is high-frequency fluctuations. This type of phenomenon can be observed at the endpoints of patients' upper limbs, where the position of the upper limb skeleton will change position at a high frequency even if there is no motion from the patient. Most importantly, this kind of fluctuation looks very similar to the motion during seizures. It will be a severe impact if the fluctuation pattern is fed into the deep learning networks.

Compared with the results obtained from testing videos, we think the joint-semiology extraction may require much higher video quality to perform as expected. The testing video has a 1080p resolution and the distance between the camera and the person is close. The extraction algorithm on the testing video has been well-performed. Unfortunately, under the current video data provided by Alfred, it is hard to find sufficient qualitative videos. Considering all these aspects, we decided to only perform the pilot test on the optical flow dataset. All the extracted skeleton data for the joint-semiology method is properly saved for future analysis and reuse. On the other hand, the extracted features from optical flow are reasonable. These data samples will be used to perform the pilot test.

5.2 Deep Learning Model Performance on Public and Own Dataset

5.2.1 Performance on Public Dataset

The following figure shows the training accuracy on the UCF101 dataset of the 'two-stream-action-recognition' method (only motion stream). The training accuracy finally reached 82% in about 12 epochs. In the model, I have let the learning rate to auto fits the loss; in the figure, we can see that the accuracy does not vary a lot as the epochs increase. From the observation, the model does not happen to be overfit or underfit, and the loss is monotone decreasing for most of the time. The training accuracy stops increasing when the training loss decreases to 0.5.

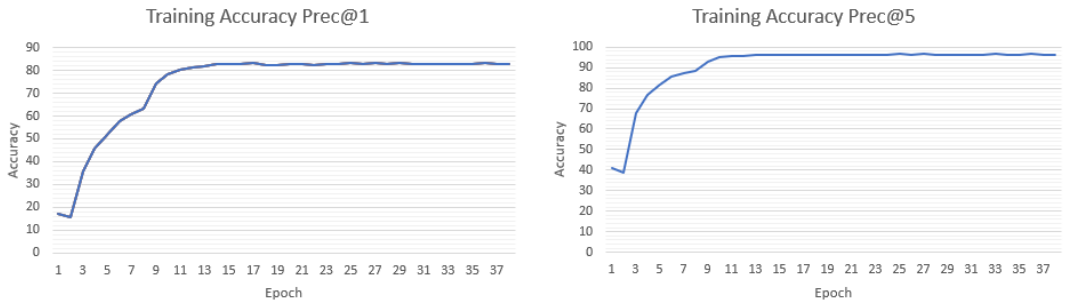


Figure [5.2.1-1] Training Accuracy for Precision @ 1 and 5

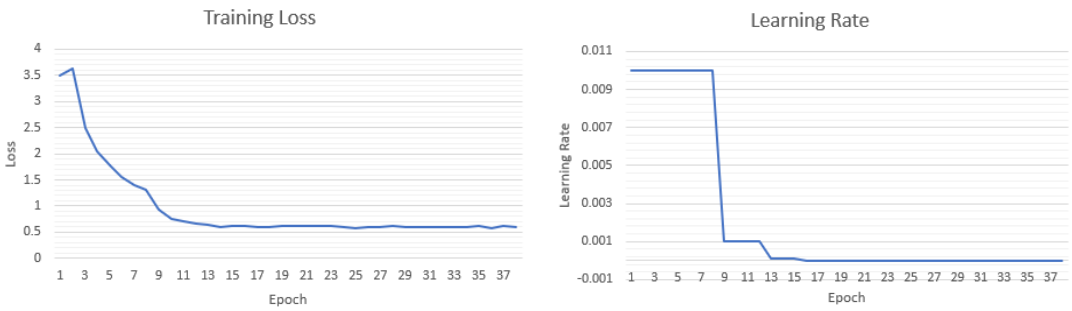


Figure [5.2.1-2] Training Loss and Learning Rate

Figure 5.2.1-3 and 5.2.1-4 show the testing accuracy of the UCF101 dataset. The accuracy for precision 1 stabled at 78% in the end. And the loss on testing is decreased to 7.

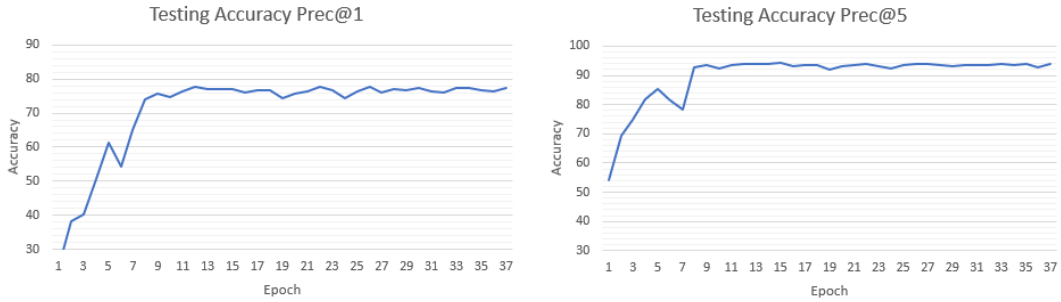


Figure [5.2.1-3] Testing Accuracy for Precision @ 1 and 5

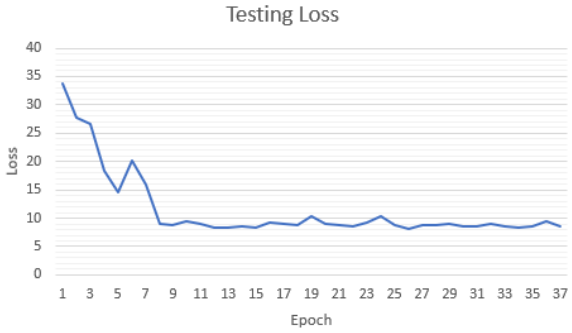


Figure [5.2.1-4] Testing Loss

Overall, the motion stream of the 'two-motion-action-recognition' model shows the expected performance as reported by [12]. All these above results are obtained from ResNet101. It is also noted the UCF101 dataset is quite large to give this performance.

The HCN model has been tested on NTU-RGB+D (public) dataset, its performance is shown in table 5.2.1-1

Table [5.2.1-1] Testing Accuracy and Loss on NTU-RGB+D Dataset

Epoch	Testing Accuracy @ Precision 1	Testing Accuracy @ Precision 5	Test Set Loss
50	72.8%	93.6%	1.193
100	76.9%	95.2%	0.988
150	79.9%	96.2%	0.870

200	81.4%	96.6%	0.819
250	82.3%	96.9%	0.756
300	82.8%	97.1%	0.723
350	83.2%	97.2%	0.710
400	83.5%	97.2%	0.697

The testing accuracy of the HCN model rises fast during the first 100 epochs, and the accuracy increasing rate starts to decrease from epoch 150~200. After 400 epochs, the testing accuracy at precision 1 is 83.5%, and the loss is 0.697. This result is matched with [13]. Although the HCN model has achieved great success in NTU-RGB+D Dataset, this model is not used in the pilot testing stage. Because the extracted skeleton in NTU's dataset has extremely high quality, all the input videos are recorded under perfect lighting conditions and the distance between the person and camera is close. As discussed in section 5.1.2, the dataset extracted from Alfred's videos does not have enough precision.

5.2.2 Pilot Test on Alfred Dataset

Note that all the accuracy/ loss shown in section 5.2.2 are the averaged results from 5 different experiments. Each of the experiments is started from scratch with different training and testing sets split. The reason to perform multiple experiments is that the Alfred dataset is too small. There are 21 samples (7 GTC seizures and 14 non-seizures) without a further split dataset (as mentioned in section 4.1.4), and 144 samples (102 GTC seizures and 42 non-seizures) with the further split. Under such a small amount of data, the accuracy of the test set will not able to reflect general performance. It will be likely that all of the easy samples are included in the test set, which results in an unreal high accuracy.

Following the discussion made in section 4.3, the comparison between the effect of performing further split, the effect of using ROI, the effect of different versions of ResNet, and the effect of using different stack sizes will be shown.

The following table shows the influence of performing further dataset split.

Table [5.2.2-1] Effect of Dataset Further Split

Conditions	Training Accuracy	Testing Accuracy	With ROI	With Further Split
Res18-Stack size 10	65.866%	56.785%	N	N
Res18-Stack size 10	69.423%	58.210%	N	Y

The following table shows the influence of using different stack sizes without further splitting datasets.

Table [5.2.2-2] Effect of Different Stack Size

Conditions	Training Accuracy	Testing Accuracy	With ROI	With Further Split
Res18-Stack size 10	65.866%	56.785%	N	N
Res18-Stack size 50	64.187%	51.750%	N	N

As results are shown in the above two tables, the training and testing accuracy for all these different conditions are quite low. With the dataset further split, the performance of the deep learning model has increased slightly. And increasing the stack size seems does not have too much impact. The main reason for low accuracy considered here is not using ROI. As discussed in previous sections, it is important to filter out irrelevant motion in the environment by applying the region of interest.
After applying the region of interest to all of the testing conditions, the following results have been obtained.

Table [5.2.2-3] Comparison between ResNet and Stack Size

Conditions	Training Accuracy	Testing Accuracy	With ROI	With Further Split
Res18-Stack Size 10	73.199%	63.727	Y	Y
Res18-Stack Size 20	73.346%	64.826%	Y	Y
Res18-Stack Size 50	69.778%	58.874%	Y	Y
Res34-Stack Size 10	73.170%	62.003%	Y	Y
Res34-Stack Size 20	74.741%	65.512%	Y	Y
Res34-Stack Size 50	67.455%	55.509%	Y	Y
Res101-Stack Size 10	71.829%	61.201%	Y	Y
Res101-Stack Size 20	70.532%	62.650%	Y	Y
Res101-Stack Size 50	68.114%	55.780%	Y	Y

Table 5.2.2-3 illustrates the model performance of using different versions of residual networks and different sizes of stacked images. It is noticed that as the stack size increases to 50, all of the models under different conditions will give a worse performance. This may be because when using 50 as stack size, the input will be a 100x224x224 image. (2 channels each stack 50 frames gives 100) The 100-dimensional image is too large for the model to learn from it. Meanwhile, the performance for stack sizes 10 and 20 is quite close, with a stack size of 20 being slightly better. The main reason behind this could be it has a better view of the time series motion. For stack size 20, there will be 20 consecutive frames stacked together, which can better reflect motion of seizures. The ResNet34 seems most suitable here, as the insufficient amount of data may underfit the model with too many parameters. Among all the different versions, ResNet34 with stack size 20 gives the best performance. The detailed training and testing accuracy, and the loss is shown in the following figure.

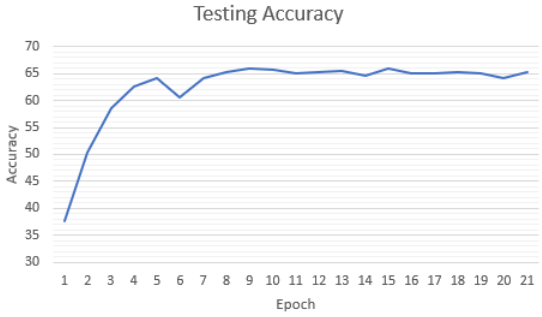


Figure [5.2.2-1] Testing Accuracy for Res34-Stack20

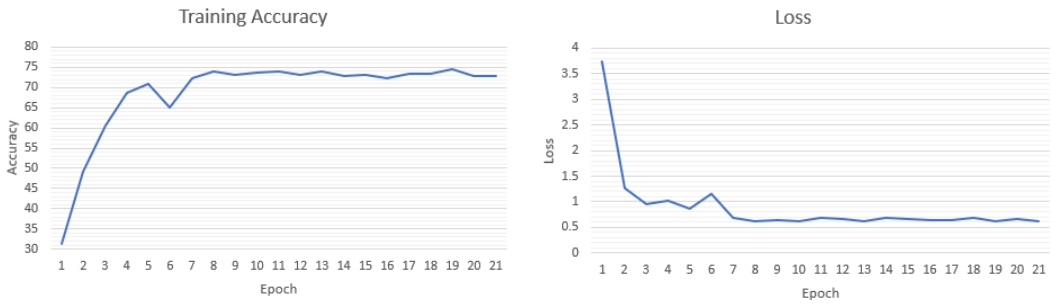


Figure [5.2.2-2] Training Accuracy and Loss for Res34-Stack20

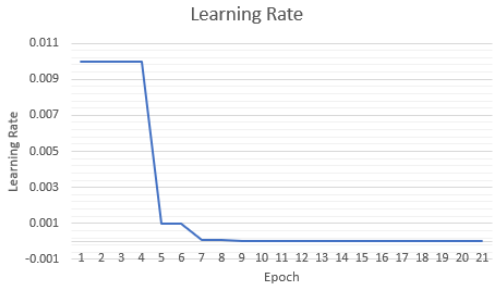


Figure [5.2.2-3] Learning Rate for Res34-Stack20

Based on Figure 5.2.2-1~5.2.2-3, the training accuracy reached 74% in 7~8 epochs, and then it stopped increasing. The loss at this time is about 0.5, which means there are almost no more features left for the model to learn. The corresponding learning rate at that time is also decreased to 1E-5, which helps the model to find its global minimum. After that, the model converged. A similar trend can also be observed in the testing accuracy plot, in which the accuracy was finally fixed at 65%.

The gap between the accuracy of the self-dataset and the UCF101 dataset mainly depends on the number of data samples. This training pipeline can be reused when there is sufficient amount of data has been provided, and the accuracy is expected to be higher.

One thing to note is Alfred did not provide the annotations for PNE seizures, therefore, the model's performance on seizures classification cannot be verified. But based on the performance of the UCF101 dataset, the accuracy in terms of seizure classification can be expected.

5.3 Challenges

There are several challenging parts to this project. Based on the timeline, the first challenge would be doing development on the Nvidia Jetson devices. Unlike common Linux Operating Systems, the Nvidia Jetson cannot directly install some dependencies easily. Many versions of dependencies are even not available to be running on the device. Therefore, some simple tasks on Nvidia Jetson devices would normally require a lot of searches. Meanwhile, setting up the Nvidia Jetson devices from scratch is also challenging, during this process, I learned many commands in Linux and did a lot of searches on the developer forum. Another challenge in this project would be the data privacy concern. As mentioned, I do not have access to view the

original video input, therefore, it is hard for me to get an intuition of the data extraction performance. However, this is a valuable experience for me to work on data that don’t have full access to. At the deep learning model training and testing stage, the challenges are the insufficient amount of dataset. This directly influenced the final performance of the model. Unfortunately, there is no solution to fix this issue, as the data set annotation job can be only done by Alfred. Some common deflections in the extracted dataset are also a challenging part for this project. There are different technics, for instance the ROI mechanism, has been implemented to reduce the deflections.

6 Conclusion and Future Works

In this project, a pipeline for extracting privacy-preserved data has been successfully implemented and deployed in the hospital. Using the pilot data extracted, we have performed several experiments with different conditions, and finally performed the seizures detection. The pipeline and pilot data collected in this project will be valuable for this topic to dig deeper in the future.

At the current stage, dataset insufficiency (only 9 GTCS videos have been provided) is the primary constraint for gaining a high accuracy. The feature extraction process will keep operating in the Alfred Hospital to generate more data.
And more choice of deep learning networks can be tested and evaluated.

7. Communication

The communication used in this project is mainly via Ding Talk. The advantage of using that is we can perform instant communication, typically, the response time is less than 1 hours. This has brought a lot of convenience for both the student and the supervisors.
The project meetings in held via Zoom, where everyone can get suggestions from peer and supervisors.

References

[1] "Epilepsy", *Who.int*, 2022. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/epilepsy. [Accessed: 04- May- 2022].

[2] “The Truth about Psychogenic Nonepileptic Seizures,” Epilepsy Foundation. [Online]. Available: https://www.epilepsy.com/stories/truth-about-psychogenic-nonepileptic-seizures

[3] "Alfred Health | Improving the lives of our patients | Alfred Health", *Alfred Health*, 2022. [Online]. Available: https://www.alfredhealth.org.au/. [Accessed: 05- May- 2022].

[4] "Monash Medical AI Group", *Monash Medical AI Group*, 2022. [Online]. Available: https://www.monash.edu/mmai-group. [Accessed: 05- May- 2022].

[5]. David Ahmedt-Aristizabal, Clinton Fookes, et al. Aberrant epileptic seizure identification: A computer vision perspective Seizure: European Journal of Epilepsy 65 (2019) 65–71.

[6]. Evelien E. Geertsema, et al. Automated video-based detection of nocturnal convulsive seizures in a residential care setting Journal. 2018, 59(S1):53

[7]. "Why Deep Learning over Traditional Machine Learning?", *Medium*, 2022. [Online]. Available: https://towardsdatascience.com/why-deep-learning-is-needed-over-traditional-machine-learning-1b6a99177063. [Accessed: 09- May- 2022].

[8]. David Ahmedt-Aristizabal, Kien Nguyen, et al. Deep Motion Analysis for Epileptic Seizure Classification IEEE publication. 2018, P2-3.

[9]. D. Ahmedt-Aristizabal, K. Nguyen, S. Denman, S. Sridharan, S. Dionisio and C. Fookes, *"Deep Motion Analysis for Epileptic Seizure Classification,"* 2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 2018, pp. 3578-3581, DOI: 10.1109/EMBC.2018.8513031.

[10] "Deploy AI-Powered Autonomous Machines at Scale", *NVIDIA*, 2022. [Online]. Available: https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-agx-xavier/. [Accessed: 10- May- 2022].

[11] "GitHub - Daniil-Osokin/lightweight-human-pose-estimation-3d-demo.pytorch: Real-time 3D multi-person pose estimation demo in PyTorch. OpenVINO backend can be used for fast inference on CPU.", *GitHub*, 2022. [Online]. Available: https://github.com/Daniil-Osokin/lightweight-human-pose-estimation-3d-demo.pytorch. [Accessed: 11- May- 2022].

[12] Yi Huang, Rajat Shrivastava “two-stream-action-recognition: Using two stream architecture to implement a classic action recognition method on UCF101 dataset", *GitHub*,

2022. [Online]. Available: https://github.com/jeffreyyihuang/two-stream-action-recognition. [Accessed: 17- May- 2022].

[13] "GitHub - huguyuehuhu/HCN-pytorch: A pytorch reproduction of { Co-occurrence Feature Learning from Skeleton Data for Action Recognition and Detection with Hierarchical Aggregation }.", *GitHub*, 2022. [Online]. Available: https://github.com/huguyuehuhu/HCN-pytorch. [Accessed: 17- May- 2022].

[14] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition", *arXiv.org*, 2022. [Online]. Available: https://arxiv.org/abs/1512.03385. [Accessed: 18- May- 2022].

Appendix 1

Code used in this project can be found in this repository
https://github.com/1Zhaoning/FYP-repository.git

Contact: luzhaoning@outlook.com, zluu0020@student.monash.edu