INTERACTIVE SEMANTIC SEGMENTATION SOFTWARE FOR ASSISTING MEDICAL DOCTORS IN ASSESSING PATIENT'S CONDITIONS FROM IMAGES AND VIDEO

โปรแกรมเพื่อช่วยแพทย์ในการวิเคราะห์และวินิจฉัยการรักษาโรคโดยใช้เทคโนโลยีการแบ่งส่วนวีดีโอและรูปภาพทางการแพทย์

BY

MR. TEERAWIT KUNVARANGKUL 5988142
MR. JIRAYUT SAENGSIWARIT 5988200
MR. CHANON UDOMTAWEEDECH 5988260

ADVISOR
DR. AKARA SUPRATAK

CO-ADVISOR
DR. THANAPON NORASET

A Senior Project Submitted in Partial Fulfillment of the Requirements for

THE DEGREE OF BACHELOR OF SCIENCE (INFORMATION AND COMMUNICATION TECHNOLOGY)

Faculty of Information and Communication Technology
Mahidol University
2019

ACKNOWLEDGEMENTS

This research has become a reality with the kind support and help of many individuals. We would like to extend our sincere thanks to all of them. Foremost to our senior project advisor, Dr. Akara Supratak, who always gave us constant encouragement, useful guidance, and considerable supervision in every stage of our research. His willingness to give this time so generously has been very much appreciated.

Besides our advisor, we would like to express our special gratitude to our senior project co-advisor, Dr. Thanapon Noraset, who always gave us clarification and constructive suggestions during the planning and development of this research work. This made our project completed successfully. Furthermore, we owe deep gratitude to our senior project committees: Aj. Sanit Sanghlao and Dr. Thanapon Noraset for their insightful comments and suggestions throughout this project.

Last but not the least, we also grateful to all instructors and the Faculty of Information and Communication Technology, Mahidol University for providing the knowledge and resources to be used in this project. The project could not have been this successful without all of the aforementioned support.

Mr. Teerawit Kunvarangkul
Mr. Jirayut Saengsiwarit
Mr. Chanon Udomtaweedech

INTERACTIVE SEMANTIC SEGMENTATION SOFTWARE FOR ASSISTING MEDICAL DOCTORS IN ASSESSING PATIENT'S CONDITIONS FROM IMAGES AND VIDEO.

MR. TEERAWIT KUNVARANGKUL 5988142 ITCS/B
MR. JIRAYUT SAENGSIWARIT 5988200 ITCS/B
MR. CHANON UDOMTAWEEDECH 5988260 ITCS/B

B.Sc. (INFORMATION AND COMMUNICATION TECHNOLOGY)

PROJECT ADVISOR: DR. AKARA SUPRATAK

ABSTRACT

Most of the early stages of diagnoses for defining the right therapy, medical doctors often assess from the observation of bodily disorders and movement disorders. Both are the main factors for medical doctors to identify the symptom at the beginning stage. However, there is plenty of limitation that prevents them from diagnosing the causes of disease thoroughly. For example, the limited numbers of medical doctors that insufficient to effectively take care of a large number of patients, the responsibilities that medical doctors have to perform, or even the deficiency in modern equipment that can help to reduce time and increase capability in the diagnosis process. Currently, there are much software has been developed to support medical doctors in the diagnosis process. But that software also has some complexity to install and use, including the lack of accuracy caused by the low number of the data training set. The team has a goal to develop the system to help medical doctors diagnosing the diseases and plan the treatment process by using semantic segmentation. The software can identify the locations of each component (e.g., the upper eye ages) in the photos or videos, tracking movements, and read values for analyses to determine the abnormal movements. Medical doctors can use the result to diagnose the symptom and defining the right therapy.

KEYWORDS : OBJECT DETECTION, SEMANTIC SEGMENTATION, MACHINE LEARNING, TRAIN MODEL, MEDICAL DIAGNOSIS

35 P.

โปรแกรมเพื่อช่วยแพทย์ในการวิเคราะห์และวินิจฉัยการรักษาโรคโดยใช้เทคโนโลยีการแบ่งส่วนวีดีโอและรูปภาพทางการแพทย์

ธีรวิทย์ กุลวรางกูร 5988142 ITCS/B
จิรายุส แสงศิวะฤทธิ์ 5988200 ITCS/B
ชานน อุดมทวีเดช 5988260 ITCS/B

วท.บ. (เทคโนโลยีสารสนเทศและการสื่อสาร)

อาจารย์ที่ปรึกษาโครงการ: ดร. อัคร สุประทักษ์

บทคัดย่อ

พื้นฐานของการวินิจฉัยและวางแผนการรักษาโรคส่วนใหญ่ บุคลากรทางการแพทย์มักจะประเมินจากการสังเกตุความผิดปกติของร่างกายหรือความผิดปกติของการเคลื่อนไหว ที่ถือเป็นปัจจัยสำคัญสำหรับบุคลากรทางการแพทย์ที่ใช้วินิจฉัยโรคในเบื้องต้น แต่เนื่องด้วยข้อจำกัดหลายประการที่ถือเป็นปัจจัยสำคัญไม่ว่าจะเป็น จำนวนบุคลากรทางการแพทย์ที่ไม่สัมพันธ์กับปริมาณของผู้ป่วย จำนวนภาระหน้าที่ที่ต้องรับผิดชอบ หรือ การขาดอุปกรณ์ที่มีประสิทธิภาพที่สามารถช่วยบุคลากรทางการแพทย์สามารถทำการวิเคราะห์และวินิจฉัยโรคได้อย่างรวดเร็วและแม่นยำ ปัจจุบันได้มีการพัฒนาซอฟต์แวร์ต่าง ที่มีเป้าหมายเพื่อช่วยในการวินิจฉัยโรคของบุคลากรทางการแพทย์ แต่ซอฟต์แวร์เหล่านั้นก็มีอุปสรรคบางประการ ไม่ว่าจะเป็นวิธีการหรือขั้นตอนในการใช้งานที่มีความยุ่งยากและซับซ้อน หรือขาดจำนวนของชุดข้อมูลสำหรับใช้ในการเรียนรู้ที่มีปริมาณไม่เพียงพอ ทีมพัฒนาจึงได้พัฒนาระบบที่ช่วยบุคลากรทางการแพทย์ในการวิเคราะห์และวินิจฉัยการรักษาโรคโดยใช้เทคโนโลยีการแบ่งส่วนวีดีโอและรูปภาพทางการแพทย์ โดยโปรแกรมสามารถตรวจจับการเคลื่อนไหวของอวัยวะเป้าหมายและวิเคราะห์ค่าตัวเลขในรูปแบบต่าง ๆ เพื่อบุคลากรทางการแพทย์สามารถนำข้อมูลเหล่านี้ ไปใช้ประกอบการวินิจฉัยโรคและมอบแนวทางการรักษาให้แก่คนไข้

35 หน้า

CONTENTS

CONTENTS (Cont.)

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

LIST OF FIGURES (Cont.)

Page

CHAPTER 1
INTRODUCTION

This chapter comprises five sections, which are Motivation, Problem Statement, Objectives of the Project, Scope of the Project, and Expected Benefits.

1.1 Motivation

Precise diagnosis is essential for defining the right therapy at an early stage. However, many cases are very difficult to identify the symptoms at the beginning. Moreover, the limited numbers of medical doctors also increase the duties that doctors will be given. Not only the time pressures and duties of the medical doctors having fatigues but the variances of the data used by them for the diagnoses partially prevent them to thoroughly diagnose the causes of diseases [1]. In addition, most diagnosis tasks involve visual inspection with the patient's body in order to find the abnormal symptoms. From all previous problems, It's inspired our team to develop software that analyzes the visual images or videos in order to help medical doctors diagnose and plan treatments for patients accurately and quickly.

Machine learning is a method of optimizing the performance criterion using past experience from datasets. In addition, machine learning can determine a certain symptom or a health risk assessment much faster with higher accuracy. Most of the time, machine learning mainly used in medical diagnosis for making critical decisions, as the data in the medical field is huge, and the accuracy of the diagnosis depends on considering the huge data of the patients. Therefore we would like to apply the experiment of machine learning to improve the accuracy of the diagnosis of the disease.

1.2 Problem Statement

Currently, some software uses for diagnosing the symptom such as Emotric, and a Real-time 3D Eyelids Tracking model [2] [3]. However, the existing software has some limitations about the datasets that make the system cannot automatically identify the area of interest in order to make measurements for diagnosis. For example, Emotrics, doctors have to manually annotate the area of interest with a dot-line in every single frame of a video to make the system can calculate measurements. In addition, a similar problem from both software is doctors cannot provide feedback or make a correction to a model. In detail, when doctors found fault results, they have to record that information and report to programmers in order to make a correcting. By reasons that doctors don't have the know-how and experiences on machine learning to make a correcting by themselves.

Developing machine learning is challenging because we have a problem with limited datasets. The problem of data scarcity is very important since data are at the core of any machine learning models. The size of a dataset is often responsible for poor performances in machine learning models [4]. In addition, datasets for training the machine learning models must be provided by experts with real understandings of the data [5]. To make machine learning have a decision as accurate as done by the experts. A system that helps medical doctors create a quality dataset and use a machine learning model without machine learning expertise is necessary. Since there is no existing software that sufficient enough for general people, who don't have know-how experiences in machine learning, develop or train machine learning models with themselves. It's required to develop systems that support those groups of people to help them gather quality datasets and create powerful machine learning by themselves.

1.3 Objectives of the Project

1.3.1 To help medical doctors identify the area of interest by using the Semantic Segmentation technique to reduce time-consuming in the diagnosis process and follow the treatment result.

1.3.2 To develop a web application that allows doctors to annotate videos or images for training semantic segmentation models and view the output, including increase datasets to train a model.

1.4 Scope of the Project

This project starts with a pilot study by developing a system that helps the ophthalmologists. They can use the Semantic Segmentation System to analyze the blink rate from a slow-motion video. In addition, they can teach Machine Learning models to detect the lower and upper eyelids by redrawing the incorrect eye edges. The detected eye image is essential information that can be used to diagnose the effectiveness of neurological and fatigue disorders, such as Parkinson's disease and closed-eye disease (Blepharospasm) through the web application.

1.5 Expected Benefits

1.5.1 Doctors can correct the mistakes which are made by the semantic segmentation model easily without machine learning knowledge and generate more training examples in order to improve model performance.

1.5.2 Doctors can use the semantic segmentation model to reduce time-consuming and increase more convenient in the diagnosis process.

1.5.3 Doctors can utilize the semantic segmentation technique for analyzing medical photos and videos to diagnosis the symptoms and to promptly follow treatment results.

CHAPTER 2
BACKGROUND

This section contains two mains sections, which are Background and Literature reviews.

2.1 **Background**

2.1.1 Object Detection

Object detection is a technique used to detect various objects into corresponding categories on the image. This technique has input as the image that contains several objects inside, and the output of this technique consists of 5 main values of each object which is detected. The first pair of the output value is the position of the object in the x and y-axis. The second pair of output is the width and height of the object. The final output is the class of the object, such as human, dog, cat, or tree. For example, find a certain class in the image as Figure 2.1, which contains objects such as dog, bicycle, and truck. Then the program will generate the output in the format as x, y, width, height, and a class of the object.

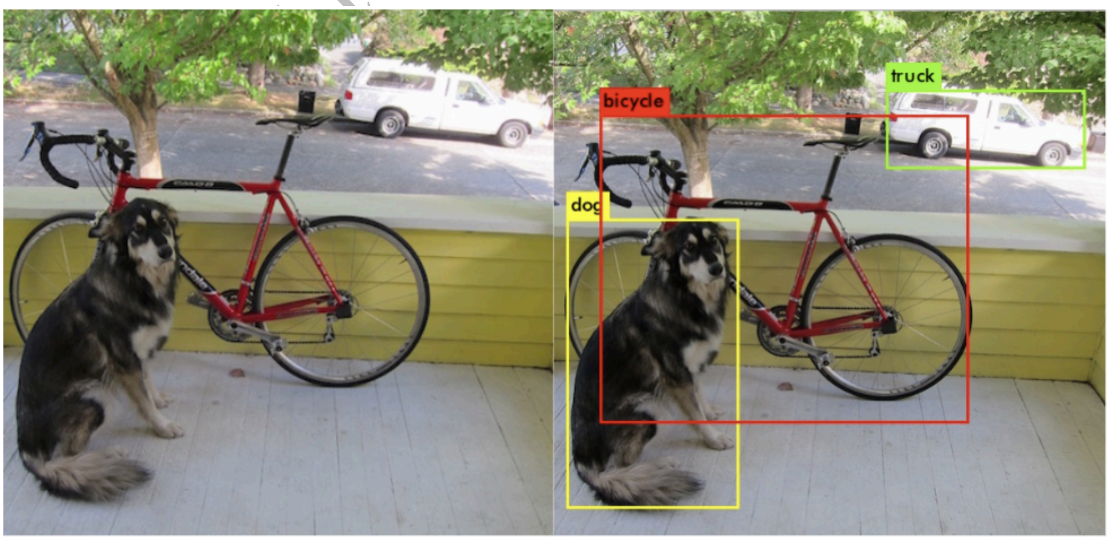


Figure 2.1: An example of image detection [6].

2.1.2 Semantic Segmentation

Semantic Segmentation is the technique used to extract essential information on an image such as the number of objects, the position of objects, and the shapes of objects. One of this technique uses to determine an edge pixel which must be near to the boundary between an object and the background, or between two objects [7]. This is an example of using semantic segmentation to extract objects (car, tree, pole, and people) shown in Figure 2.2 below.

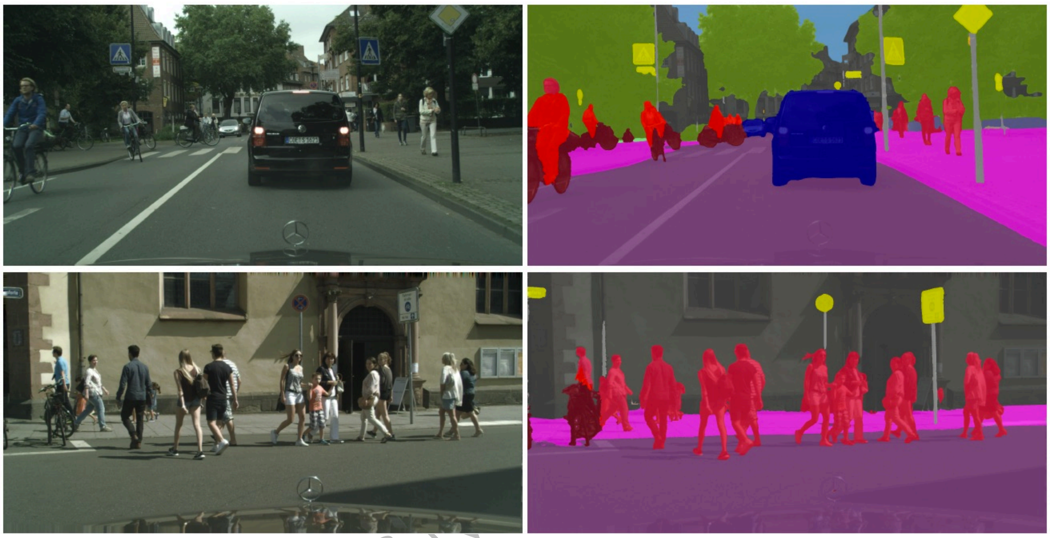


Figure 2.2: An example of Image Segmentation [8]

In the medical environment, the semantic segmentation technique is used to simplifying and changing the display to help analyze in medical diagnosis. Segmentation also helps to make things simple and transform the representation of medical images into the images that can be used for analysis. For instance, the area of cancer is checked with a clearer picture of cancer results by quality analysis [9], as shown in Figure 2.3 [10]. Therefore, medical professionals can measure various values from images quickly and can use it to improve the efficiency in the diagnosis of the disease.

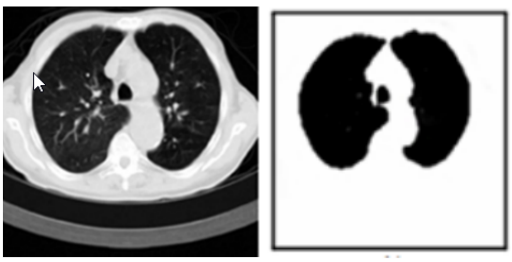


Figure 2.3: An example of X-ray lungs (using image segmentation) [11]

2.1.2.1 Health issues related to eyes

Eye health can signal vision problems or other diseases, such as diabetes or stress. Moreover, people can see for themselves or come to check with doctors. By the way, the single best way to protect eye vision is a medical examination with ophthalmologists

2.1.2.2 The symptoms of Facial Nerve Palsy

Facial Nerve Palsy is one type of neurologic disorder that occurs in the patient suddenly. In Facial Nerve Palsy, patients cannot move their upper and lower parts of their faces on one side. As one of the causes, the patient cannot move or blink the eye normally, such as Figure 2.4 below.

Figure 2.4: An example of Facial Nerve Palsy patient [12]

2.1.2.3 The symptoms of Dry Eye Disease

Dry Eye disease is a common ocular surface disease characterized. For example, the loss of tear film homeostasis is evaporation-induced tear film hyperosmolarity, which damages the ocular surface by inflammation, leading to the disease [13].

2.1.2.4 The symptoms of Blepharoptosis

Blepharoptosis is also a common ocular disease, it is characterized by an abnormal drooping of the upper eyelid, and affected individuals have difficulty lifting their eyelids. For example, according to Figure 2.5, the patient with involutional ptosis of both upper eyelids. The left upper eyelid is significantly more ptosis. On the left side are an elevated eyelid crease, superior sulcus deformity, and arched an eyebrow [14].

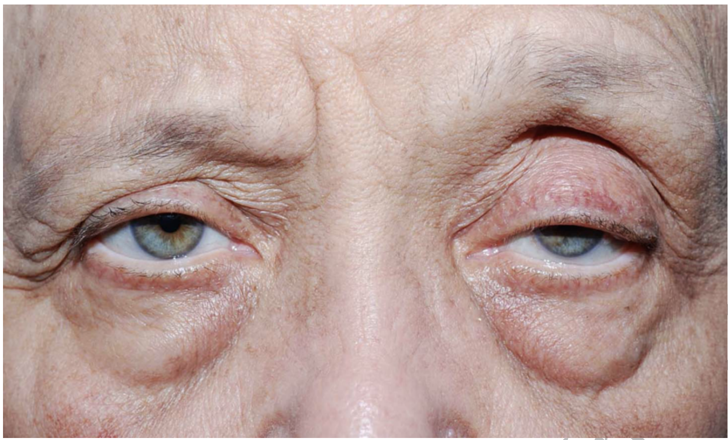


Figure 2.5: An example of a patient's Blepharoptosis disease [14].

2.1.2.5 Measurements to diagnose healthy eyes

There are several measurements which use to diagnose healthy eyes. The first measurement is the blink duration and blinks velocity. These parameters calculated from the start of upper eyelid descent until maximum recovery in each isolated blink in ptosis and the control eye blink [15]. Doctors use blink duration and blink velocity to diagnose the symptoms of facial nerve palsy because patients are difficult to blink or cannot blink. After treatment, the doctor will examine this symptom by checking patients' blink. Another measurement is the blink frequency. It is a wide range of blink frequencies. It was observed during testing through a slow-motion camera. The sequential series of images represent more information that can be used to analyze eye blink [16]. The doctors use blink frequency to observe the symptoms of Dry Eye disease because patients have unusual blinking behavior. It means patients blink a few times until the eyes become dry. After treatment, doctors must calculate the number of patients' blink by blink frequency. The last measurement is the palpebral aperture. It is the vertical distance between the central points of the upper and lower eyelid margins. Doctors use palpebral aperture to diagnose the symptoms of Blepharoptosis because the patient has abnormal drooping of the upper eyelid [15]. Doctors will examine the patient's disease from the upper and lower eyelid margins.

However, there are some weak points after the doctor treated patients by different doctors. Firstly, doctors cannot specify the actual result of treating. This is because it is difficult to get accurate data and the objective measurements of each disease. Another weak point is a subjective diagnosis of doctors. It means each doctor gives different opinions to patients on diagnosing the disease.

2.2 **Literature Reviews**

2.2.1 Existing software and techniques

2.2.1.1 Example of Medical technology use image to diagnosis

Emotrics is an automated facial measurement in Facial Palsy. Emotrics is an existing software using image-processing techniques to support the part of calculated facial measurements [2]. Currently, it is used by doctors in the clinical environments for facial measurements as Figure 2.6. Even though this program provides a function to calculate measurements, Emotrics cannot provide user-friendly experiences. Doctors have to annotate a point by point as landmarks on the patient image, which is labor-intensive and time-consuming. Emotrics can support only images; therefore, it's not possible to diagnose symptoms about movement parameters such as blink rate and blink velocity.

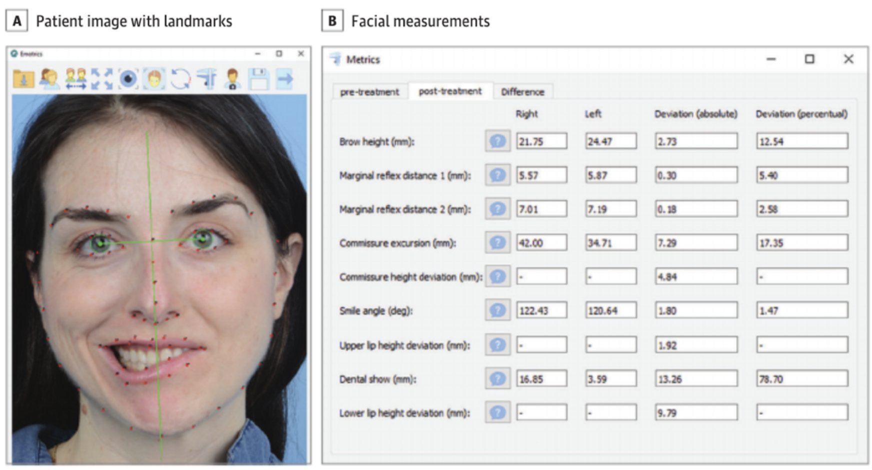


Figure 2.6: An example of Emotrics' GUI and Facial measurements(B) Computed from the Patient image with Facial Landmarks(A) [2].

2.2.2 Research papers

2.2.2.1 Real-time 3D Eyelids Tracking from Semantic Edges

They create a program for a face and eye-tracking system to obtain full face results with more detailed eye regions by using image segmentation as Figure 2.7 [3]. They manually create an annotated dataset in order to train the model to label face and eye edges. The technique which we can learn in this research paper is that we can use the segmentation techniques to improve our project in the part of the label eyes' edges. Therefore, the problem is no more dataset, and it is manually labeled.

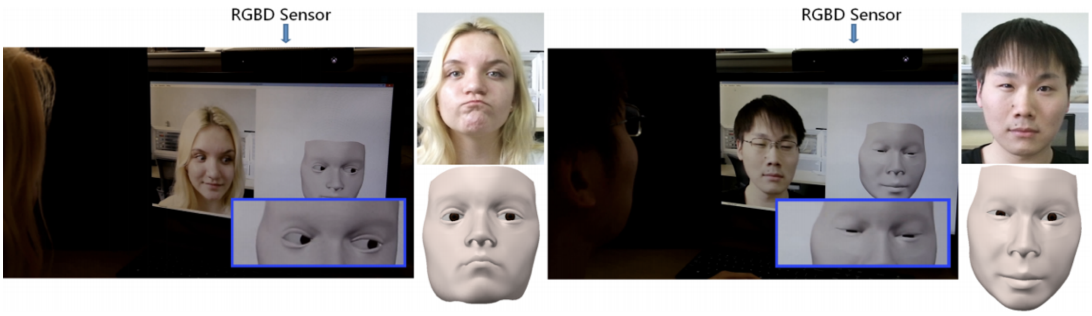


Figure 2.7: An example of Real-time 3D Eyelids Tracking from Semantic Edges Program [3].

CHAPTER 3
METHODOLOGY

In order to achieve our objective, we develop a system that can create a model that can annotate a video from a slow-motion camera. Then we will use an annotated video train a machine learning model. Due to a high frame rate of slow-motion cameras that faster than 240 frames per second [17], it's enough to measure human motions such as blink frequency, and blink velocity. So, we will develop a web application that doctors can use to annotate a video for training the Semantic Segmentation model.

3.1 Web-Interface Flow

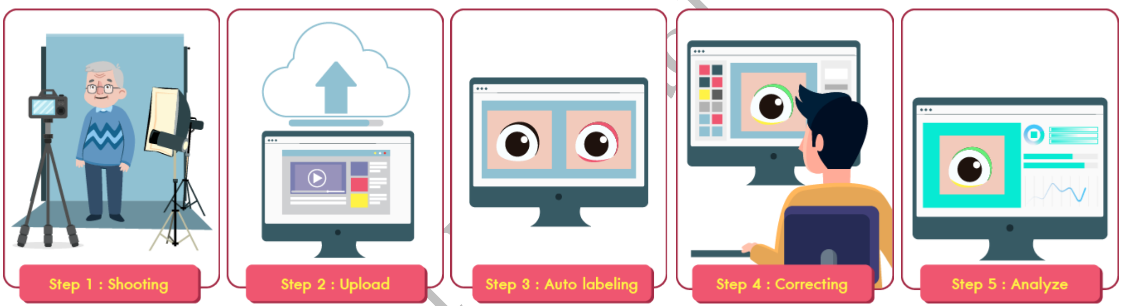


Figure 3.1: Web-Interface Flow

The system interface flow above shows how doctors can use the system. The process starts with doctors recording a patient's video by using a slow-motion camera (Step 1) and uploads the video into the system through the web interface (Step 2). The system will automatically identify the area of interest in every single frame. Then, the annotated image will appear on the screen (Step 3). If the system displays the result that does not match with the doctors' needs, doctors can edit and reannotate each frame again in order to train a machine model and increase accuracy (Step 4). Finally, doctors can see movement analysis and use it to diagnose the symptoms (Step 5).

3.1.1 Step 1: Recording the slow-motion video

Doctors have to record a slow-motion video with a frame rate of 120-240 fps. The way to recording video is simple, start with a straight face recording, and the patient has to look at the camera through the recorded during. The recording time should be around 30 seconds in slow-motion mode and distance between the patient and the recording camera should not over 60 cm since it might affect the resolution after we upload it to the system.

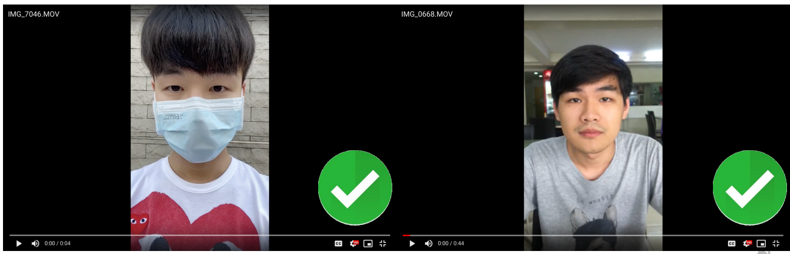


Figure 3.2: Example of a camera's angle.

3.1.2 Step 2: Upload the slow-motion video to the system

Doctors have to upload the recorded video to the system, and the recorded file should be in .mov, .mp4, .avi, a4v file extension.

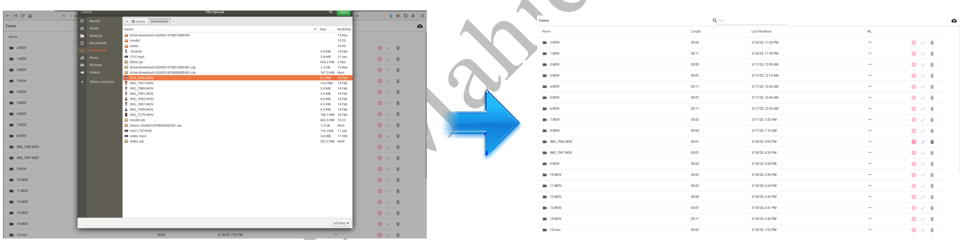


Figure 3.3: Uploading a video interface. The first image file selection interface (left). The second image is the uploaded video list (right).

3.1.3 Step 3: Auto labeling

The main interface will show the human-face, left eye, and right eye, including the set of extract frames. Then, doctors can consider using auto labeling by using the Semantic Segmentation feature as Step 3.1.3 or manual labeling the eye edge as Step 3.1.4. If doctors select Label eyes' edges automatically, semantic segmentation inside a server will automatically identify the area of interest in every single frame. Then, the annotated image will appear on the screen.

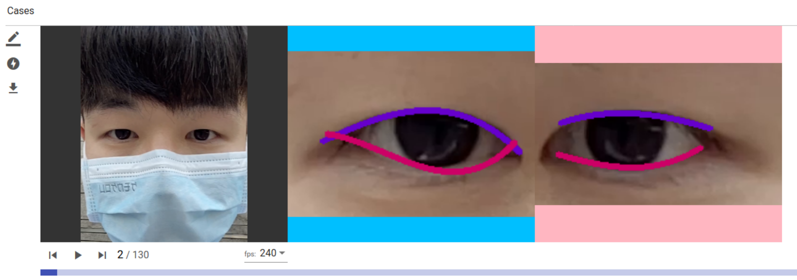


Figure 3.4: Example of auto-labeling.

3.1.4 Step 4: Make the correcting and train the machine learning model

If the application displays the result that does not match with the doctors' needs, they can make the correction by using the provided tools to relabel the incorrect eye edge, as shown in Figure 3.5. Consequently, the new annotated frame will be sent to a server in order to train a machine model. Doctors can check the model performance once again as Step 3.1.3, or even correct the model's result as Step 3.1.4.

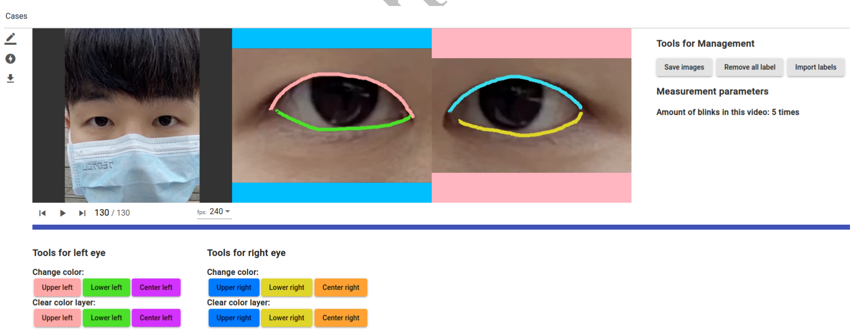


Figure 3.5: Web interface that uses to correcting models' results.

3.1.5 Step 5: Use the data the diagnosis the symptoms

The figure above shows some new values that doctors can get from our system, such as the ability to blink, and blink velocity. They can use those values to identify the system, including creating the treatment plan or even following the treatment results.

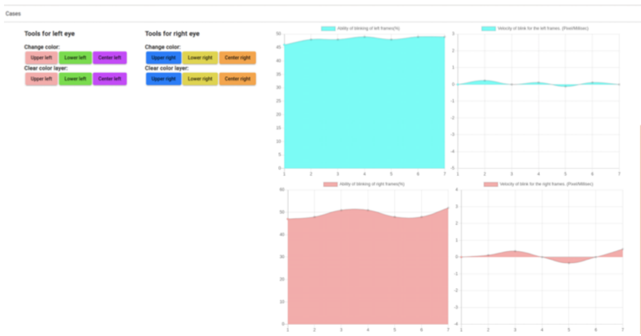


Figure 3.6: Example of movement analysis.

3.2 System Architecture

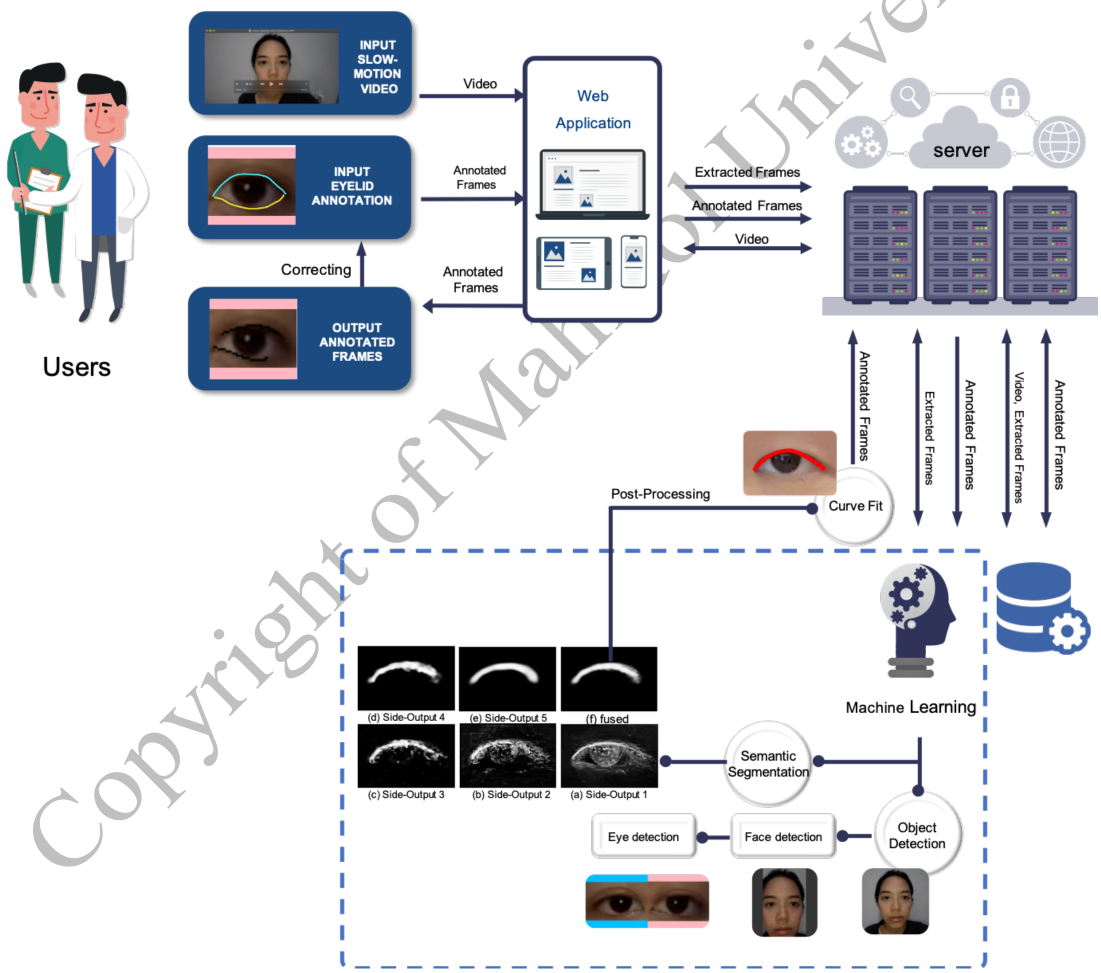


Figure 3.7: System Architecture

The system architecture above represents the process inside the system consisting of five main features. Starting with receiving video from users and sent to a server to tailor the video into frames by using FFmpeg. Then, Object Detection will detect a human face, left eye, and right eye in every single frame by using YOLOv3 and crop them. The face image will show on the screen, but the left eye and the right eye images will forward to the Semantic Segmentation model to annotate the eyelids before showing the final result on the screen. The new annotated image done by doctors will send to the semantic segmentation model directly to retrain and increase the accuracy of the model.

3.2.1 Extract frames

After doctors upload the slow-motion video to the system, the video will be sent through to the server to trailer the video into frames by using FFmpeg [18]. FFmpeg is a collection of the library tools used to process multimedia content such as video, audio, subtitles, related metadata, and widely used for image processing transcoding such as video to images (extract frames) [19] [20].

3.2.2 Extract faces and eyes

After the system gets a slow-motion video from users. The server will use YOLO V3 to detect the human face, left eye, and right eye. YOLO is one of the object detection algorithms that is widely used to detect the target component [6]. By using YOLO, we can detect multiple objects within an image. After YOLO V3 detects the target components (the human face, left eye, and right eye), the system will crop each component and show them on the screen as Figure 3.8. At the same time, the cropped eye images will send through to the Semantic Segmentation model for the annotating process.

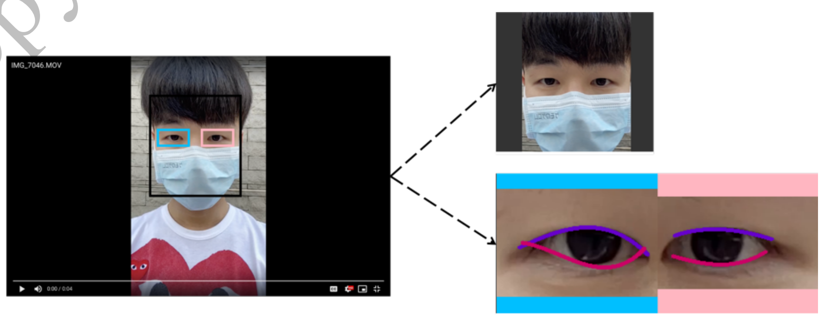


Figure 3.8: Object Detection Flow. The first image is face detection (left). The second image is eye detection (right).

3.2.3 Auto labeling (Drawing eye edges)

We apply the Semantic Segmentation technique we learned from Real-time 3D Eyelids from chapter 2. This technique will be used to label the cropped images that we get from Object Detection. The system will label both lower and upper eyelids'.

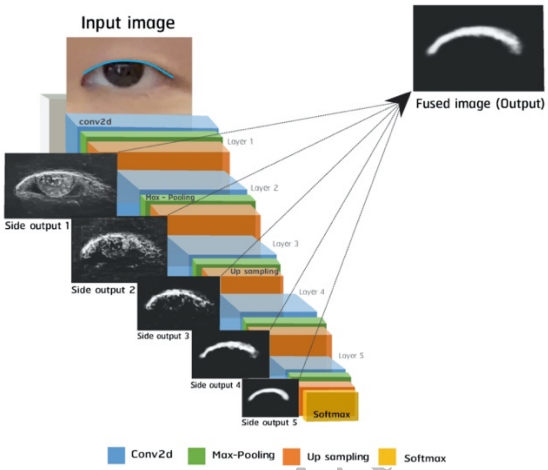


Figure 3.9: Semantic Segmentation's Architecture

The figure above shows the architecture of the Semantic Segmentation that uses the idea of a convolutional neural network. Start from receiving an input, which is the annotation of the eye image. Slide-output layers are the results of the segmentation process with eye images. The resulting from the previous layer will be the guideline improvement for the next layer in terms of increasing the accuracy of the segmentation model. Apart from using the side-output layer to be the guideline improvement, a label image done by doctors also applies in this stage to train the machine learning model. As a consequence, in the deeper side-output layer, the result will be more accurate, and irrelevant components will reduce. In the final stage of segmentation, the system will combine all side-output and apply a label image before creating the final result, which is the fused image.

3.2.4 Post-processing

In the pre-processing, we use the segmentation technique to detect the eye edges. We found that the final output from Segmentation contains a group of scattered pixels that cannot identify an actual pixel to calculate the movement values. In some cases, the segmentation gives us an incomplete result as Figure 3.10 (left). Therefore we consider using the curve fitting for post-processing to solve this problem and create the complete segmentation.

Curve fitting is the process of specifying the model that provides the best fit to the specific curves in the eye edge [21]. The fitted line plot below (Figure 3.10 (center)) illustrates the result of using a polynomial curve fitting to fit a curved relationship proportionately. As a result, we can get an actual pixel to detect the movement of each component, as shown in Figure 3.10 (right).

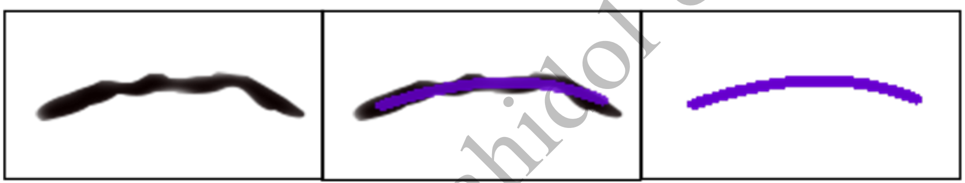


Figure 3.10: Example of post-processing. The first image is the original output from the model (left). The second image shows when using the linear relationship to fit the cure line of the original output (center). The last image is the final output with curve fitting (right).

3.2.5 Online Model Training

By the time doctors click on the train model button, the new label image from doctors will be sent through to the server for retraining the semantic segmentation model with the current model, which is used in the server. Then, the system will get the new semantic segmentation model that uses the new label images as the data sets in order to increase the performance machine labeling in Step 2.3 (Drawing eye edge or Auto labeling). Moreover, the new model that we received from the training, the doctor's data sets are also used for other videos. Consequently, the system will update the annotated result in Step 3.2.3 (Drawing eye edge or Auto labeling) and parameter result in Step 3.2.6 (Blink Dynamics Estimation).

3.2.6 Blink Dynamics Estimation

3.2.6.1 Blink distance

The first parameter which doctors can use this application to measure is the blink distance between the upper and lower eyelids. We calculate from the highest pixel of the upper eyelid edge and lowest pixel of the lower eyelid edge in each extracted frame with the formula.

$$D = \sqrt{(x2 - x1)^2 + (y2 - y1)^2}$$

Let (x1, y1) be a variable point of the highest pixel of the upper eyelid and let (x2, y2) be a variable point of the lowest pixel of the lower eyelid.

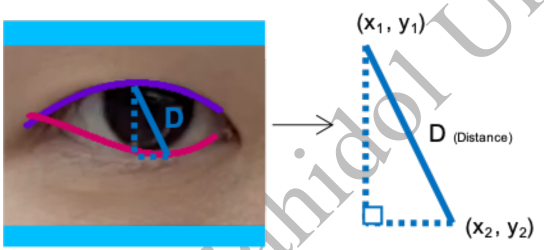


Figure 3.11: Eye edges' distance calculation

Therefore, we can measure the distance between the upper and lower eyelid edges on each frame and convert them to the percentage unit. This parameter can represent the ability of a patient's blink.

3.2.6.2 Blink velocity

The second parameter that we measure is the blink velocity. We used the result of blink distance in each frame to calculate the blink velocity according to the velocity formula.

$$v = \frac{S}{t} \quad , when\ S = D_1 - D_2$$

Let V is the velocity of blink, S is the difference between the distance from the first frame and the distance from the second frame, and t is the total time which is used to blink.

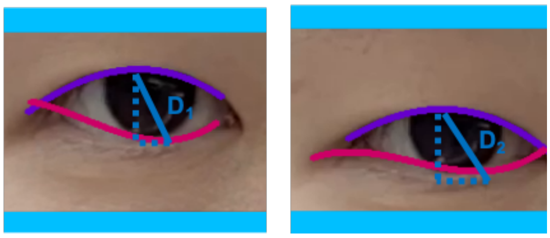


Figure 3.12: The distance from the first frame and the second frame.

3.2.6.3 Number of Blinks

The last parameter is the number of blinks. We define a value that we used to be the criteria that the algorithm will increase the number of blinking. If the distance between upper and lower eyelid edges is lower or equal to this value, it means that the patient was blinking in that frame of video.

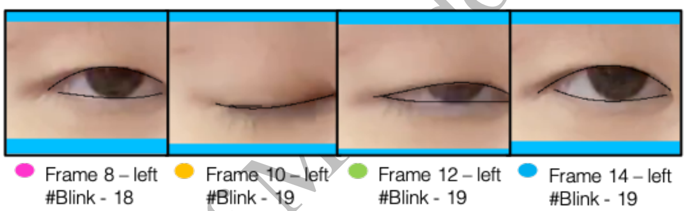


Figure 3.13: Example of blink's counting.

3.2.7 Tools we used to implement semantic segmentation and object detection.

3.2.7.1 **Python** is used to compile Python language and to stimulate Python Server in order to stimulate a server for our application.

3.2.7.2 **Angular Framework** is an additional library for Mobile Application. In addition, this framework will make our system more effective.

3.2.7.3 **YOLO** is real-time object detection. We apply YOLO with our system to detect a human-face, left eyes, and right eyes in images. Then, the detected image will be cropped to use for segmentation.

3.2.7.4 **TensorFlow** is one of the Python libraries for numerical computation. We apply this library to semantic segmentation to make machine learning faster and easier.

3.2.7.5 **PaperJS** is a plugin that provides a drawing ability in HTML5, and we use PaperJS to label the target object.

3.2.7.6 **SciPy** is one of Python libraries used for scientific computing, and technical computing. SciPy contains modules for optimization, linear algebra, integration, interpolation, special functions, FFT, signal and image processing. We use SciPy for processing in machine learning.

CHAPTER 4
EXPERIMENTAL RESULT

The experimental results are based on the two following objectives:

1. Doctors can use the web application to annotate videos or images for training semantic segmentation model and view the output, including increased datasets to train a model.
2. Doctors can use the system to automatically identify areas of interest (lower and upper eye edges), and measure the value for diagnosing the symptom correctly

4.1 **Data Collection**

4.1.1 Slow-Motion Video

4.1.1.1 Tools: iPhone Xs Pro with slow-motion feature (120 fps)

4.1.1.2 Participant: People with an age range between 20-56 with a different kind of skin and gender. We recorded 20 videos with 12 different participants.

4.2 **Evaluation**

4.2.1 Performance measurement

4.2.1.1 Mean IoU

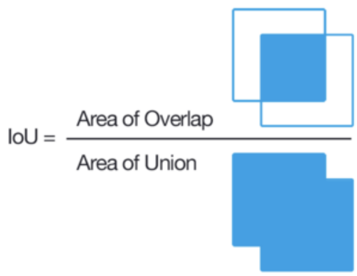


Figure 4.1: IoU formula [22]

$$IoU = Area\ of\ Overlap\ /\ Area\ of\ Union$$

We calculate mean IoU by using a number of intersections between pixels in annotating images (done by machine learning) and pixels in the label image (done by a medical doctor). Then take the intersection value divide with a number of unions between pixels in annotating images and pixels in the label image. Thereafter, we will calculate the average in IoU value and get the percent, which represents the accuracy of the model.

4.2.2 Experimental results

Table 4.1: Number of images in our training set, which separate from the Training set, Validation set, and Testing set.

	Training Set						Validation Set						Testing Set					
	Eye Image		Total	Label Image		Total	Eye Image		Total	Label Image		Total	Eye Image		Total	Label Image		Total
	Right	Left		Right	Left		Right	Left		Right	Left		Right	Left		Right	Left	
No Training	0	0	0	0	0	0	0	0	0	0	0	0	80	80	160	160	160	320
Training with 9 videos	112	112	224	224	225	448	32	32	64	64	64	128	80	80	160	160	160	320
Training with 15 videos	208	208	416	416	416	832	32	32	64	64	64	128	80	80	160	160	160	320

Our training set is based on 20 videos that contain 636 eye images and 1,268 eye edge label images, as shown in Table 4.1 above. As Table 4.2 below, we measure the similarity between label image and predictive image into mean IoU and mean Dice scores. There are 3 models and 2 conditions, which are the Indicators for accuracy and ability of the machine. First of all, No Training means we use the initial model of Paper 3D eyelid [R] for segment 5 video testing. Another model is Training with 9 videos, which separate between 13 videos of the training set and 2 videos for validation set, for segment 5 videos testing. The last model is Training with 15 videos, which separate between 13 videos of the training set and 2 videos for validation set, for segment 5 videos for testing. The first condition is using a model for segment videos with the fused image, which means the output from the model before post-processing. Another condition is using a model for segment videos with curve-fitting for post-processing.

4.2.2.1 Mean Dice

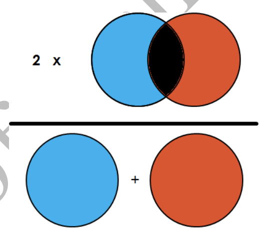


Figure 4.2: Dice formula [22]

Dice = (2 Area of Overlap) / (Total Pixel Combined)*

The mean Dice was calculated by using two multiples with a number of intersections between pixels in annotating images and pixels in the label image divided by the total number of pixels in both image.

Table 4.2: Experimental result 1 which is the performance of mean IoU and Mean Dice for upper and lower eye edge.

	Upper Eye Edge		Lower Eye Edge	
	Mean IoU	Mean Dice	Mean IoU	Mean Dice
No training	0.24 ± 0.08	0.38 ± 0.11	0.12 ± 0.03	0.22 ± 0.06
No training + curve fitting	0.29 ± 0.10	0.44 ± 0.13	0.16 ± 0.07	0.27 ± 0.10
Training with 9 videos	0.35 ± 0.16	0.50 ± 0.19	0.34 ± 0.12	0.34 ± 0.14
Training with 9 videos + curve fitting	0.39 ± 0.16	0.54 ± 0.19	0.36 ± 0.11	0.52 ± 0.13
Training with 15 videos	0.47 ± 0.04	0.64 ± 0.03	0.44 ± 0.03	0.61 + 0.03
Training with 15 videos + curve fitting	0.49 ± 0.04	0.66 ± 0.04	0.45 ± 0.33	0.62 ± 0.03

In our experimental result, we calculate the experimental result by 5 videos testing. We also record the mean IoU and Dice scores to measure the accuracy of the segmentation. IoU and Dice can calculate from the area (pixel) of the predictive image and the area of the annotated image. The high result will represent the high accuracy of the segmentation model. Table 4.2 shows that these numbers increase following the performance of the model according to the quantity of Training set. Both of them evaluated from "mean±SD" of an average of mean IoU and mean Dice of upper and lower eye edge of 5 testing videos with the best results from training our dataset which are 9 videos and 15 videos. There are increased 0.11, 0.12, 0.22 and 0.12, respectively for Training with 9 videos and 0.23, 0.26, 0.32 and 0.39, respectively for Training with 15 videos. Moreover, the result of using curve-fitting is increased 0.1, 0.1, 0.2 and 0.25, respectively for Training with 9 videos + curve-fitting and 0.2, 0.22, 0.29 and

0.35, respectively for Training with 15 videos + curve-fitting. Therefore, our models have good accuracy in predicting accuracy after increasing the number of data sets. It can capture the trend of the eye edges, and the curve-fitting can help improve the final result as the output image from Table 4.2.

Table 4.1: Performance of counting eyes blink.

Video_Name	No Training + Curve Fitting			Training with 15 Videos + Curve Fitting		
	Human	Machine	Error	Human	Machine	Error
Test Video 1	4	23	19	4	4	0
Test Video 2	6	11	5	6	7	1
Test Video 3	6	2	4	6	5	1
Test Video 4	14	14	0	14	17	3
Test Video 5	5	5	0	5	6	0

Table 4.3 shows that the experimental results contain 3 values, including a number of blink by human, the number of blink by machine learning, and the number of blink errors following 5 video testing. Therefore, the mean error result is 0.8 for No Training + curve-fit and 0.14 for training with 15 videos + curve-fit.

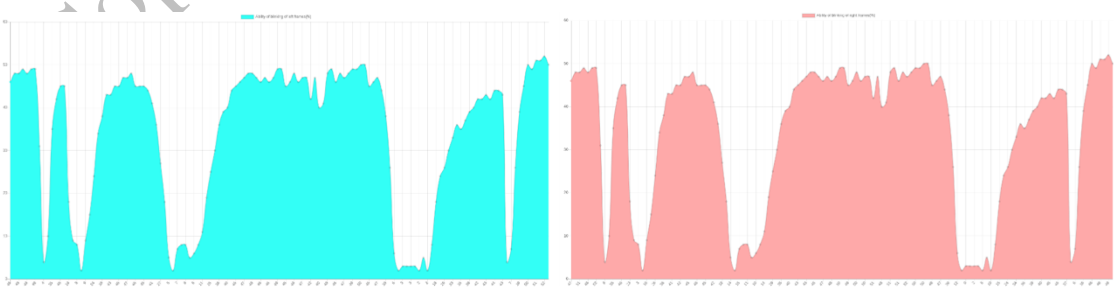


Figure 4.3: The blink pattern plotting by calculating from a distance between the upper & lower eyelids

In Figure 4.3, these graphs represent the distance between the upper and lower eyelid edges, which the patient can blink in each frame. Doctors can use this graph to compare the performance of treatment before and after the patient got the operation.

CHAPTER 5
DISCUSSION

We developed Semantic Segmentation model training by using a web-based application for medical doctors. They can use our system to increase the performance of the Semantic Segmentation model without know-how in machine learning developing experience. In addition, they can use our system to diagnose the symptom by measuring the patient's movement values. However, after we complete the developing stage, we test our system with the tiny number of the data set, and we receive the result, as shown in Chapter 4. The final output has the accuracy enough to measure the movement values and diagnosis of the disease. However, in the developing and evaluation process, we also found some problems and we apply some solutions as follows:

5.1 Even the results from semantic segmentation will contain a group of scattered pixels that affect the accuracy, as shown in Figure 5.1. However, we apply the curve fitting that can solve this problem by constructing the annotated line. As a result, the accuracy was increased, as shown in the mIoU Table 4.2. The result from curve fitting can be used to detect the eye edges with high precision and measure the movement value correctly. In addition, Semantic Segmentation also helps medical doctors get the new value that can be used to increase the efficiency in the diagnosis process.

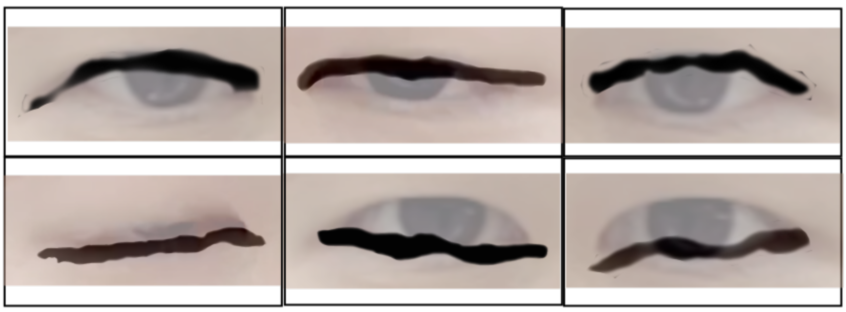


Figure 5.1: The example of Semantic Segmentation's outputs.

5.2 The resolution of eye images can affect the performance of machine learning. If the data set has a low resolution of images, the model will get low accuracy that is computed from mean IoU and mean Dice. This is because the thickness of the predictive label on the eye edges depends on the resolution of images, as shown in Figure 5.2 below. However, each eye image in both videos is recorded with the same camera. Therefore, the distance between the camera and patient directly affects the accuracy of our system. In detail, if the distance between both two objects is farther away (over 60 cm.), the cropped eye images will get a low resolution. This problem affects our model when we use the model to segment eye edge labels, including calculating the fault parameters in Step 2.6 (Blink Dynamics Estimation).

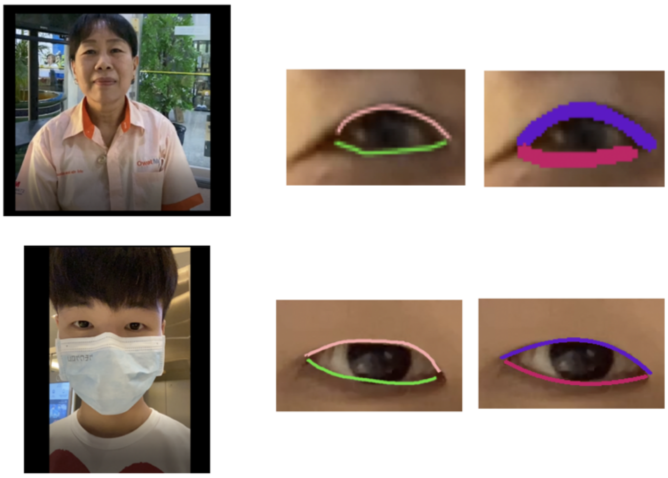
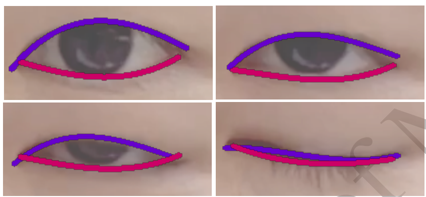


Figure 5.2: The example result of different resolution video. Compare with low (66 x 67 pixels) (top row) and high (225 x 135 pixels) (bottom row) resolution of eye images from two videos.

5.3 The performance that shows through mean IoU and mean Dice scores is increased following the performance of the model according to the quantity of Training set. If we put more datasets in the training set, we will get a better model than the old model. For example, in Table 4.2, the performance of training 15 videos (416 eye images / 832 eye edge label images) model has more performance than Training

9 videos (224 eye images / 448 eye edge label images) and No Training (No our dataset).

5.4 According to Chapter4, mean IoU and mean Dice get low values that are unreasonable with the final output of the model. As an example, in Figure 5.3, the output from machine learning has high accuracy when observed with the naked eyes. However, when we take IoU and Dice to measure the performance, the result wasn't what we expected. However, we try to use another way to evaluate our model's performance, which is the error in the clinical-related metrics, such as the number of blinks, and blink velocity. The result we got from machine learning is accurate when compared with the result from doctors. In detail, the system can do the objective measurement (e.g., the number of blinks, blink velocity, etc.) correctly when compared with the measurement from the nacked eye. In the future, we need to study and find better measurement metrics to measure the performance of Semantic Segmentation that more suitable for our experiment.



	Upper Eye Edge	Lower Eye Edge
Mean IoU	0.39 ± 0.16	0.36 ± 0.11
Mean Dice	0.54 ± 0.19	0.52 ± 0.13

Figure 5.3: Example of the final output (first image) compared with mean IoU and mean Dice score (second image).

CHAPTER 6
CONCLUSION

This chapter summarizes what has been desired and what the achievement in the project which connects a brief summary of the project, the implementation, the evaluation results, and future work.

We propose the semantic segmentation model training system that achieves to use to assess the patients' abnormalities. Due to the low number of datasets related patient, medical professionals are impelling to generate the quality dataset that can use for progressively advanced diagnosis for patients.

In developing the model training system, we apply the web-based application system with a convolutional neural network to demonstrate the performance of the semantic segmentation system to medical doctors, including allowing them to build-up the model performance. We also apply the polynomial curve fitting to improve the final result from the segmentation before appearing on the screen.

However, we cannot develop our system as real-time processing because we use the slow-motion video to do the experiments. The high number of frames is the main problem that generates a long time processing problem. Besides, in the practical, we necessary to record the video with hight frame rates because we want to capture all human movement for the diagnosis.

We evaluate the performance of our model using 20 videos that contain 636 pair of eye images. Our model can detect eye edges with the mean IoU and mean Dice of 0.49±0.04, 0.66±0.04 consequently from the upper eye edge and 0.45±0.03, 0.62±0.03 therefore from the lower eye edge. Based on this performance, our model can accurately count the number of blinks with a mean error of 0.14 percent. Although the mean IoU and mean Dice is small, it still annotated eye eadges correctly. The curve-fitting can use to improve the model result efficiency and can be used in the clinical environment.

In the future, we might use other metrics that can reflect the true performance of our model, such as using the Overlap Confident [23] that can provide the better measurements between the ground truth and the predictive image. Also, we might develop to measure many kinds of movement values as doctors' needed for diagnosis. Besides, this system can expand to detect another organ for making a comprehensive diagnosis.

REFERENCES

[1] B. BORRELL, "The effects of long work hours on medical residents," Los Angles Time, 11 October 2010. [Online]. Available: https://www.latimes.com/archives/la-xpm-2010-oct-11-la-he-pro-con-work-hours-20101011-story.html. [Accessed 2 November 2019].

[2] J. D. Diego Luis Guarin, "A Machine Learning Approach for Automated Facial Measurements in Facial Palsy," JAMA Facial Plastic Surgery , March 2018. [Online]. Available: https://www.researchgate.net/publication/323784094_A_Machine_Learning_Approach_for_Automated_Facial_Measurements_in_Facial_Palsy. [Accessed 14 November 2019].

[3] F. X. M. L. a. J.-H. Y. T. U. QUAN WEN, "Real-time 3D Eyelids Tracking from Semantic Edges," September 2017.

[4] A. Gonfalonieri, "5 Ways to Deal with the Lack of Data in Machine Learning," KDnuggets, [Online]. Available: https://www.kdnuggets.com/2019/06/5-ways-lack-data-machine-learning.html. [Accessed 4 November 2019].

[5] C.Nicholson, "DatasetsandMachineLearning," 2019. [Online]. Available: https://skymind.ai/wiki/datasets-ml.. [Accessed 2 November 2019].

[6] "YOLO," 2019. [Online]. Available: https://pjreddie.com/darknet/yolo/. [Accessed 12 October 2019].

[7] T. Dwivedi, "Semantic Segmentation — Popular Architectures," towardsdatascience, 28 March 2018. [Online]. Available: https://towardsdatascience.com/semantic-segmentation-popular-architectures-dff0a75f39d0. [Accessed 2 November 2019].

[8] Stacy, "What Machine Learning Can Do," 2018. [Online]. Available: https://semanti.ca/blog/?what-machine-learning-can-do. [Accessed October 25 2019].

[9] ณ. เนียมทอง, "Image Processing กับประโยชน์ทางการแพทย์," คลังความรู้ SciMath, 23 July 2018. [Online]. Available: https://www.scimath.org/article-technology/item/7864-image-processing. [Accessed 8 November 2019].

[10] "Image Segmentation Techniques for Healthcare Systems," Hindawi, [Online]. Available: https://www.hindawi.com/journals/jhe/si/480657/cfp/. [Accessed 2 November 2019].

[11] ณ. เนียมทอง, "Image Processing กับประโยชน์ทางการแพทย์," 23 July 2018. [Online]. Available: https://www.scimath.org/article-technology/item/7864-image-processing. [Accessed November 12 2019].

[12] M. W. C. M. C. Michael, "Facial Nerve Palsy," MSD MANUAL Professtional Version, June 2019. [Online]. Available: https://www.msdmanuals.com/professional/neurologic-disorders/neuro-ophthalmologic-and-cranial-nerve-disorders/facial-nerve-palsy. [Accessed 12 November 2019].

[13] Y. Su, Q. Liang, G. Su, N. Wang, C. Baudouin and A. Labbé, "Spontaneous Eye Blink Patterns in Dry Eye: Clinical Correlations," IVOS an ARVO journal, October 2018. [Online]. Available: https://iovs.arvojournals.org/article.aspx?articleid=2711429. [Accessed 2 November 2019].

[14] P. B. M. Lakshman Subbaraman, "The Tear Film: More Complex Than You May Realize," REVIEW OF CORNEA & CONTACT LENSES, 15 February 2018. [Online]. Available: https://reviewofcontactlenses.com/article/the-tear-film-more-complex-than-you-may-realize. [Accessed 14 November 2019].

[15] A. H. K.-A. K. M. E. G. E. R. F. M. a. D. G. E. Felix H. W. Mak, The Royal Society Publishing, 12 March 2016. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4843668/. [Accessed 14 November 2019].

[16] J.-M. S. J.-M. H. Woon-Hee Lee, "The analysis of eye blinking pattern using high-frame-rate camera.," PubMed.gov, July 2017. [Online]. Available: https://www.ncbi.nlm.nih.gov/pubmed/29060166. [Accessed 14 November 2019].

[17] "Slow motion," [Online]. Available: https://en.wikipedia.org/wiki/Slow_motion. [Accessed 2 November 2019].

[18] "https://www.ffmpeg.org/," [Online]. [Accessed 12 January 2020].

[19] Richardpl, "GitHub," [Online]. Available: https://github.com/FFmpeg/FFmpeg. [Accessed 18 December 2019].

[20] "Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/FFmpeg. [Accessed 18 December 2019].

[21] J. Frost, "Curve Fitting using Linear and Nonlinear Regression," Statistic By Jim, [Online]. Available: https://statisticsbyjim.com/regression/curve-fitting-linear-nonlinear-regression/. [Accessed 28 January 2020].

[22] E. Tiu, "Metrics to Evaluate your Semantic Segmentation Model," towards data science, 10 August 2019. [Online]. Available: https://towardsdatascience.com/metrics-to-evaluate-your-semantic-segmentation-model-6bcb99639aa2. [Accessed 18 December 2019].

[23] B. Rees, "Similarity in graphs: Jaccard versus the Overlap Coefficient," 4 May 2019. [Online]. Available: https://medium.com/rapids-ai/similarity-in-graphs-jaccard-versus-the-overlap-coefficient-610e083b877d. [Accessed 20 May 2020].

[24] B. BORRELL, "The effects of long work hours on medical residents," Los Angeles Times, 11 October 2010. [Online]. Available: https://www.latimes.com/archives/la-xpm-2010-oct-11-la-he-pro-con-work-hours-20101011-story.html. [Accessed 2 November 2019].

[25] S. P. V. F. R. Benenson, "Large-scale interactive object segmentation with human annotators," 2019.

[26] R. Benenson, "object segmentation with human annotators," 2019.

BIOGRAPHIES

NAME	Mr. Teerawit Kumvarangkul
DATE OF BIRTH	08 April 1998
PLACE OF BIRTH	Phuket, Thailand
INSTITUTIONS ATTENDED	Phuketwittayalai School, 2017 Hight School Diploma Mahidol University, 2020 Bachelor of Science (ICT)
NAME	Mr. Jirayut Saengsiwarit
DATE OF BIRTH	06 November 1997
PLACE OF BIRTH	Rayong, Thailand
INSTITUTIONS ATTENDED	Assumption College, 2017 Hight School Diploma Mahidol University, 2020 Bachelor of Science (ICT)
NAME	Mr. Chanon Udomtaweedech
DATE OF BIRTH	15 May 1998
PLACE OF BIRTH	Prachinburi, Thailand
INSTITUTIONS ATTENDED	Prachinratsadorn-amrong School, 2017 Hight School Diploma Mahidol University, 2020 Bachelor of Science (ICT)