RASPBERRY PI BASED MOBILE SYSTEM FOR TONGUE DIAGNOSIS WITH A U-NET SEGMENTATION AND K-MEANS CLASSIFICATION

AHMED MOHAMED GAAFAR MAHMOUD ALI

UNIVERSITI TEKNOLOGI MALAYSIA

PSZ 19:16 (Pind. 1/13)

UNIVERSITI TEKNOLOGI MALAYSIA

DECLARATION OF THESIS / UNDERGRADUATE PROJECT REPORT AND COPYRIGHT

Author's full name :

Date of Birth :

Title :

Academic Session :

I declare that this thesis is classified as:

☐	**CONFIDENTIAL**	(Contains confidential information under the Official Secret Act 1972)*
☐	**RESTRICTED**	(Contains restricted information as specified by the organization where research was done)*
☐	**OPEN ACCESS**	I agree that my thesis to be published as online open access (full text)

1. I acknowledged that Universiti Teknologi Malaysia reserves the right as follows:
2. The thesis is the property of Universiti Teknologi Malaysia
3. The Library of Universiti Teknologi Malaysia has the right to make copies for the purpose of research only.
4. The Library has the right to make copies of the thesis for academic exchange.

Certified by:

SIGNATURE OF STUDENT	**SIGNATURE OF SUPERVISOR**
MATRIC NUMBER	**NAME OF SUPERVISOR**
Date:	Date:

NOTES : If the thesis is CONFIDENTIAL or RESTRICTED, please attach with the letter from the organization with period and reasons for confidentiality or restriction

“I hereby declare that I have read this final year project report and in my opinion this final year project report is sufficient in terms of scope and quality for the award of the degree of Bachelor of Electronic Systems Engineering (Engineering)”

Signature	:	______________________
Name of Supervisor	:	PROF. MADYA IR. DR. OOI CHIA YEE
Date	:	JANUARY 27, 2023

RASPBERRY PI BASED MOBILE SYSTEM FOR TONGUE DIAGNOSIS WITH A U-NET SEGMENTATION AND K-MEANS CLASSIFICATION

AHMED MOHAMED GAAFAR MAHMOUD ALI

A final year project report submitted in partial fulfilment of the
requirements for the award of the degree of
Bachelor of Electronic Systems Engineering (Engineering)

Universiti Teknologi Malaysia
Malaysia Japan International Institute of Technology
Universiti Teknologi Malaysia

JANUARY 27, 2023

DECLARATION

I declare that this final year project report entitled *"Raspberry Pi based Mobile System for Tongue Diagnosis with a U-Net Segmentation and K-Means Classification"* is the result of my own research except as cited in the references. The final year project report has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature	:	
Name	:	AHMED MOHAMED GAAFAR MAHMOUD ALI
Date	:	JANUARY 27, 2023

DEDICATION

This thesis is dedicated to my father, who taught me that the best kind of knowledge to have is that which is learned for its own sake. It is also dedicated to my mother, who taught me that even the largest task can be accomplished if it is done one step at a time.

ACKNOWLEDGEMENT

All glory to Almighty God for allowing me to start my studies and for providing me with the health and strength to begin this research. I'd like to thank my supervisor, Prof. Madya Ir. Dr. Ooi Chia Yee, for her guidance and support throughout this proposal. I would like to extend my thanks and appreciation to my co-supervisor, Dr. Choo, as well as Embedded Systems ikohza, who assisted us throughout our projects. I am also grateful to the lecturers, staff, and fellow students at Malaysia-Japan International Institute of Technology (MJIIT), Universiti Teknologi Malaysia, and everyone else who assisted me throughout my studies. Last but not least, none of this would have been possible if it were not for my parents, so special thanks to them and my family for everything they have done for me to be able to achieve this degree.

ABSTRACT

Tongue Diagnosis is a discipline of Traditional Chinese Medicine (TCM) that involves assessing and interpreting physical characteristics of the tongue in order to make a TCM-based diagnosis on a patient. However, traditional diagnosis is based on visual inspections, which is difficult to standardise since environmental factors such as shadows appearing on the tongue can affect classification accuracy. Moreover, the current segmentation algorithms are limited in terms of the environment where they can be used, as they require the image to be taken from a specialised image acquisition device with specific illumination setting which limit the application of the tongue diagnosis system to be used on other platforms. Therefore, this project developed a segmentation based on the U-Net network and health classification algorithm based on K-Means clustering for the tongue to overcome the above limitation. Moreover, this project provides a convenient platform for medical practitioners, researchers and patients which is a mobile application that allows the images to be taken from a smartphone camera to perform tongue diagnosis on a mobile platform. The trained segmentation model resulted in an accuracy of 96.37% from images which is taken from the specialised image acquisition device, and accuracy improvement from 3.75% to 76.67% for images taken from the smartphone in various environments allowing better classification results, achieved an accuracy of 76% for healthy tongues and 73% for unhealthy tongues. Overall, the developed system provides more objective measurements of assessing the tongue by a precise segmentation of the tongue, accurate classification and a convenient platform for the user.

ABSTRAK

Diagnosis Lidah ialah satu disiplin Perubatan Tradisional Cina (TCM) yang melibatkan penilaian dan tafsiran ciri fizikal lidah untuk membuat diagnosis berasaskan TCM pada pesakit. Walau bagaimanapun, diagnosis tradisional adalah berdasarkan pemeriksaan visual, yang sukar untuk diseragamkan kerana faktor persekitaran seperti bayang-bayang yang muncul pada lidah boleh menjejaskan ketepatan pengelasan. Selain itu, algoritma segmentasi semasa adalah terhad dari segi persekitaran di mana ia boleh digunakan, kerana ia memerlukan imej diambil daripada peranti pemerolehan imej khusus dengan tetapan pencahayaan khusus yang mengehadkan penggunaan sistem diagnosis lidah untuk digunakan pada platform lain. Oleh itu, projek ini membangunkan segmentasi berdasarkan rangkaian U-Net dan algoritma klasifikasi kesihatan berdasarkan pengelompokan K-Means untuk lidah untuk mengatasi had di atas. Selain itu, projek ini menyediakan platform yang mudah untuk pengamal perubatan, penyelidik dan pesakit yang merupakan aplikasi mudah alih yang membolehkan imej diambil dari kamera telefon pintar untuk melakukan diagnosis lidah pada platform mudah alih. Model pembahagian terlatih menghasilkan ketepatan 96.37% daripada imej yang diambil daripada peranti pemerolehan imej khusus dan peningkatan ketepatan daripada 3.75% kepada 76.67% untuk imej yang diambil daripada telefon pintar dalam pelbagai persekitaran yang membolehkan hasil pengelasan yang lebih baik, dicapai ketepatan 76% untuk lidah yang sihat dan 73% untuk lidah yang tidak sihat. Secara keseluruhannya, sistem yang dibangunkan menyediakan ukuran yang lebih objektif untuk menilai lidah dengan pembahagian lidah yang tepat, klasifikasi yang tepat dan platform yang mudah untuk pengguna.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

TCM	-	Traditional Chinese Medicine
CPU	-	Central Processing Unit
HSV	-	Hue, Saturation, Value
ROI	-	Region Of Interests
CNN	-	Convolutional Neural Networks
DL	-	Deep Learning
ReLU	-	Rectified Linear Unit
HDF5	-	Hierarchical Data Format 5
TIAS	-	Tongue Image Analysing System

LIST OF SYMBOLS

γ	-	Gamma
σ	-	Sigma
ε	-	Var Epsilon

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Research Background

1.1.1 Tongue Diagnosis and TCM

Tongue Diagnosis is an aspect of Traditional Chinese Medicine (TCM) that involves evaluating and interpreting physical characteristics of the tongue in order to perform a TCM-based diagnosis of a patient. It is a non-invasive, effective method of performing an auxiliary diagnosis at any time and in any location, which can help to meet the global need in the primary healthcare system. This method of diagnosis has long been used by traditional medicine practitioners in both East Asian and Western countries (Anastasi *et al.*, 2019). Tongue Diagnosis dates back to the Shang Dynasty (1600 - 1046 B.C.) and consists of visually inspecting the tongue body for vitality, colour, shape, moisture, and movement, followed by assessing the tongue coating for colour, thickness, distribution, and characteristics at the root (Kirschbaum, 2000).

The tongue is an organ that reflects the body's physiological and clinicopathological conditions. Each part of the tongue corresponds to a different internal organ. The condition of the tongue is thought to indicate the essence of the patients' disorders. Physicians visually assess the colour, shape, and coating of the tongue and use the results to make treatment decisions. According to (Jung *et al.*, 2012), analysing the tongue colour provides useful diagnostic information such as blood congestion, water imbalance, and psychological problems.

1.1.2 Automated Tongue Diagnosis Systems

Most of the time, medical doctors rely solely on just observations, their knowledge, and experience when examining a tongue for TCM-based diagnosis which may lead to errors due to environmental factors such as shadows appearing on the tongue causing the colour to change, which could be avoided with using tools and the recent automation advanced technologies. Automated tongue diagnosis is a system that aids medical doctors to standardise different procedures and render reliable diagnosis in order to enhance the clinical application value of Chinese Medicine. They are computer-based systems which take tongue images from any input, camera or a stored digital image; and then capture the characteristics of the tongue being captured. These systems are also provided with backups capabilities and means to store the tongue data or it also has the advantage to be used with advanced medical recording systems that help practitioners to save the patients' health records for easy access in the future (Tania *et al.*, 2019; Lo *et al.*, 2009).

1.1.3 Segmentation algorithms

Tongue image segmentation falls under the category of medical image segmentation. Medical image segmentation is the extraction of regions of interests (ROIs) from 3D image data, such as MRI or CT scans. The primary goal of segmenting this data is to identify areas of the anatomy that are needed for a specific study. The main objective from performing segmentation on medical images is that it allows for a more precise analysis of anatomical data by removing background objects which are not related to the diagnosis and only focusing on the necessary areas of the image (Synopsys, 2023).

There have been various applications and advancements of image segmentation by incorporating Artificial Intelligence (AI), to achieve more effective and accurate results. As shown in (Moeskops, 2022), various methods of implementing an image segmentation algorithm includes a contour-based approach which searches for the edges of the ROI, a voxel-based segmentation which uses thresholding to identify the ROI, registration-based methods which segments based on a sample of a dataset that will generate an output with similar patterns as it was in the input. Deep learning has been used in the segmentation algorithms as well, with various algorithms including fully convolutional networks (FCNs), U-net and Dilated convolutional network which are all examples of Convolutional Neural Networks (CNNs).

1.1.4 IoT Utilisation in healthcare

The Internet of Things (IoT) is an emerging technology that uses the cloud to connect smart devices with sensors. Its goal is to collect and process data while also sharing it among devices. This enables the devices to generate meaningful output. In the field of medicine. IoT devices can be used to remotely monitor patients, allowing for earlier detection of diseases and for faster interventions. IoT can provide clinicians with more information to help them to make better decisions by collecting data from a variety of sources. Patients can use IoT devices to track and share their own health data with their healthcare providers. This will increase the convenience of medical inspections and improve communication between patients and doctors. IoT can help to reduce healthcare costs by eliminating the need for traditional methods such as face-to-face consultations (Ismail, 2019).

1.2 Problem Statement

Although traditional tongue colour diagnosis is an effective method for assessing patients' health, its accuracy could be affected by environmental factors and the practitioners' experience. For example, when extraneous features such as shadows appear on the patient's tongue, this poses a challenge to the practitioner to judge the actual tongue colour; the varying light environment where the patient is located can also affect the tongue colour diagnosis result. Also, this traditional method is mostly done by doctors with just visual observation, description language, and empirical discrimination which may result in decreasing accuracy. As a result, an automated computer-aided tongue colour diagnosis is needed to address the shortcomings of traditional diagnosis while also improving its accuracy.

1.2.1 Limitations in segmentation algorithms

In order to develop a system which can diagnose a patient accurately and in real-time speed, there needs to be an efficient and accurate segmented tongue image with the background non-tongue elements of the body removed. Therefore, typically most computerised automated tongue diagnosis systems are composed of an image acquisition device which collects the data needed in order to perform the diagnosis, and an image segmentation algorithm that masks out non-tongue elements such as the background, face, teeth, lips, neck and other objects since the diagnosis is performed on an image which contains only the tongue body. Hence, segmentation is needed in order to produce an accurate diagnosis. However, many of the tongue segmentation algorithms take an intense amount of power resources due to the complex computation needed to be performed on each pixel on the image. Moreover, these segmentation algorithms do not perform well when the image contains a lot of noise and when the tongue is located far from the camera and only occupies a small portion of the image. Some segmentation algorithms only work with a specific dataset or with some certain lighting conditions. As shown in (Wu and Zhang, 2015), there are some difficulties in segmenting the tongue images; therefore, a smart segmentation algorithm is needed in order to remove unrelated objects from the image as an accurate tongue classification depends on the segmented tongue image. In summary, current segmentation lacks accuracy when the tongue image inputted to the system is outside the dataset considered in the algorithm, and they are limited to only being used with specialised image acquisition devices, which limits the application of computerised tongue diagnosis systems to be implemented in other platforms.

1.2.2 Platform limitations for tongue diagnosis

With the advancement of computerised tongue diagnosis systems and AI, many algorithms have been developed and integrated into the system. However, the user experience of these systems can be quite limited since it has not been integrated into an intuitive platform such as the web application or the mobile apps space. As shown in a study by (Tarute *et al.*, 2017), many of the AI applications lack the user-friendly expectances, which is limiting the potential of employing the application. Therefore, developing an intuitive platform for the user to use the automated tongue diagnosis system and utilize the AI capabilities.

1.3 Research Objective

This project aims to enhance the health quality services by developing a system that monitors the health condition of patients and address the aforementioned problems, therefore the objectives are:

1. To improve the segmentation accuracy of tongue images by employing a U-Net-based segmentation model designed specifically for medical segmentation.
2. To develop a friendly platform for users, which is a mobile application achieved by establishing communication between the smartphone and Raspberry Pi allowing for open environment tongue diagnosis.

1.4 Research Scope and Limitations

This research focuses on the development of an automated tongue diagnosis system. It will include the development of an improved U-Net based segmentation algorithm along with a friendly platform which will be a mobile application for the system. Using the labelled dataset of more than 300 images from the Kitasato university, the classification will be based on the colour of the tongue and have two classes, which will be the healthy or non-healthy patient.

In this project, the segmentation algorithm will be used on tongue images to remove the unrelated pixels from the image, such as pixels that contain the face, nose, lips, and other background objects to only have the tongue area on the image, this will allow for greater and more accurate analysis of the tongue features without the noise from the other objects.

Furthermore, a Raspberry Pi will be used as a central server that collects, processes and shares the output from the data. The data will be the tongue images which will come from the smartphone camera sensor and processing will include segmentation and classification algorithms, and it will share the output to a front-end in a mobile application.

1.5 Research Significance

This project will help to provide a reference diagnosis tool to medical doctors which will assist them to confirm the diagnosis results from multiple sources. In addition, it helps patients to perform the diagnosis remotely, hence reducing the physical contact between patients and doctors which will help in reducing the spread of infectious diseases.

The improved segmentation algorithm will aid in improving the results of the classification that are based on automated TCM and the tongue images. This will help patients, doctors, and TCM medical researchers to improve and standardises their diagnosis results. Moreover, it will provide a platform for patients and doctors that is based on a low-cost and low-power Raspberry Pi server which will make this system more convenient. The developed mobile app will help the users to manage and monitor their health status.

1.6 Thesis Outline

This thesis is outlined in five chapters including this chapter which aims to expand the tongue segmentation algorithm to be used in smartphone's camera through the mobile application and integrate with the Raspberry Pi to perform the segmentation.

Chapter 1 elaborates the background of the research, discusses the problem statement of the current tongue segmentation algorithm and the limited platform and capacity. And then, it outlined the research's objectives based on the problem statement, and describes the scope and the limitation.

Chapter 2 shows the previous work that has been done on the computerized tongue diagnosis field. The significance and reseach gap of the segmenation algorithms is discussed in detail along with the limited platform in which the tongue diagnosis could be implemented. In addition, the reseach gap and how this project is contributing to tongue diagnosis is listed.

Chapter 3 illustrates the planning and the methods applied for this research and discusses how the development environment set up is done and how the results were obtained. The methodology includes the development process of the threshold-based segmentation algorithm and the U-Net based segmenation algorithm, the process of the K-Means classification algorithm,the development Raspberry Pi and how it integrates with the mobile application, and finally the method of recording the data.

Chapter 4 Discusses the results in detail to draw a conclusion if the applied methods are valid. The results of the threshold-based and the U-Net based segmenation algorithms are compared on images taken by the smartphone's camera. In addition to the results of the applied K-Means segmentation algorithm.

Chapter 5 Outlines the conclusions of the study's findings to date and the work that needs to be done moving forward.

CHAPTER 2

LITERATURE REVIEW

2.1 Background

Many computerised tongue diagnosis systems rely on segmentation algorithms in order to automatically isolate the tongue from the rest of the mouth. This is necessary in order to accurately measure and analyse the tongue's features, which can provide valuable information about a person's health. However, many of the current segmentation algorithms lack the accuracy when the tongue image used is outside of the data set used, since most of the algorithms used for segmentation has been developed for a specific lighting setting or only considered the dataset of images with limited noise in the background.

Moreover, in the scope of this project, the current computerised tongue diagnosis systems are limiting the accessibility of the using the system. The current research is emphasising on utilising the mobile app-based health programs to improve the digital health care implementation, as shown from a study by (Lee *et al.*, 2018); it suggests that the mobile app-based are promoting the health care behaviours, and it increases the feasibility and effectiveness of health care programmes.

2.2 IoT Applications in Healthcare

IoT is a network of physical devices, vehicles, home appliances, and other items that are embedded with electronics, software, sensors, and network connectivity that enable these objects to collect and exchange data. This technology has become important and utilised in many applications nowadays. The idea behind IoT is the collection of data and processing of these data with algorithms and machine learning to produce some output.

IoT has been increasingly utilised in healthcare applications nowadays. Among them, IoT health monitoring which is the continuous monitoring of the progress of a clinical trial. Its purpose is to ensure policy, best clinical practice, regulatory norms, and standard operating procedures are followed. Monitoring is essential in a variety of medical settings. This creates an environment in which both the physician and the patient can interact. The doctor may also be aware of the current patient's health status (Pradhan *et al.*, 2022).

According to a systematic literature review by (Haghi Kashani *et al.*, 2021), IoT in healthcare can be found in a variety of medical applications. The sensor-based application includes wearable devices, and the resource-based application is related to the health IoT environment's resource heterogeneity, resource limitations, dynamic nature, and unpredictability. Also included are application-based systems such as monitoring, prediction, and detection systems, which are the focus of this project's research. The application-based domain is concerned with providing systems capable of performing one or more specific functions. In other words, in addition to sensor aspects, resource management, and communication infrastructure, an IoT-based healthcare system provides the required services to patients, carers, or users.

Deep learning is one of the processing algorithms that can be used with IoT in the application of healthcare. (Mansour *et al.*, 2021) presented a deep learning model that is developed with IoT devices. The IoT devices are used to capture the tongue images and process the images with the deep learning model, which have shown promising simulation results.

2.3 Segmentation Algorithms

(Wu and Zhang, 2015) proposed a segmentation algorithm based on combining the region and edge approaches to tongue segmentation. The region of interest is extracted as the input for the next process, and then the edge enhancement process and the generated mast of the edges are obtained. The image is then processed to extract reliable endpoints from the edge map, resulting in the segmented tongue image as shown in Figure 2.1 and the summary of the proposed flow is shown in Figure 2.2.

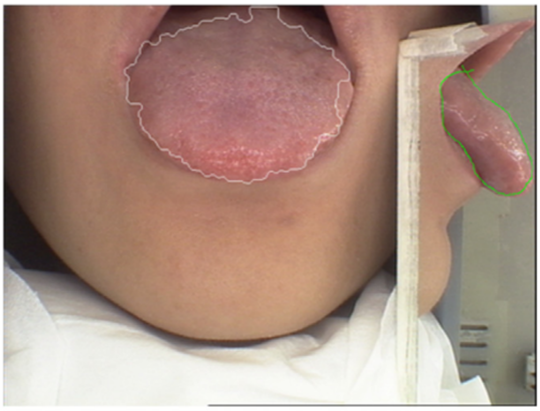


Figure 2.1 Segmentation results of (Wu and Zhang, 2015) algorithm

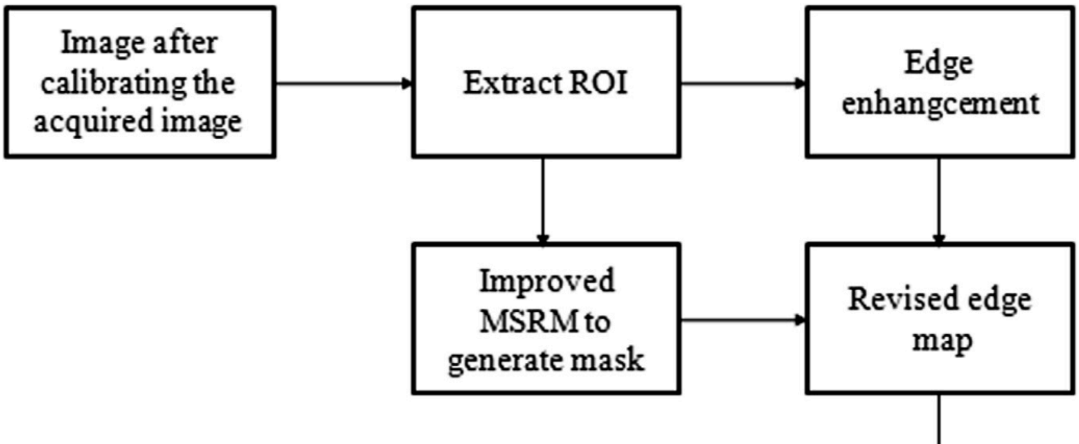


Figure 2.2 Flowchart of (Wu and Zhang, 2015) algorithm

(Li *et al.*, 2019) has shown a segmentation algorithm that is based on colour decomposition and thresholding. They proposed an efficient tongue image segmentation method after analysing the characteristics of image thresholding in different colour spaces and images. The proposed method first extracts an initial tongue body region in HSI (Hue, Saturation, and Intensity) colour space by performing image thresholding on the transformed hue component. The gap region, which is the region of interest between the tongue body root and the upper lip, is then found by adaptively selecting one of two image thresholding results on the red component of an original tongue image. Finally, to obtain the final tongue image segmentation result, the initial tongue body region is refined by removing other object regions such as the upper lip. The flow of this algorithm is shown in Figure 2.4 and the results are shown in Figure 2.3.

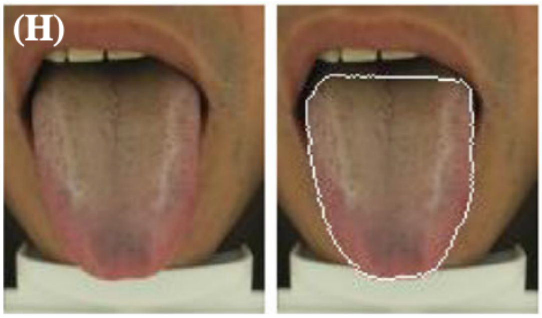


Figure 2.3 Segmentation results of (Li *et al.*, 2019) algorithm

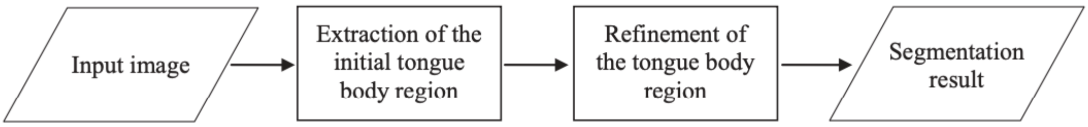


Figure 2.4 Flowchart of (Li *et al.*, 2019) algorithm

(Subandi *et al.*, 2019) also proposed a segmentation algorithm based on thresholding technique, the method starts with calculating the upper threshold value based on the concept that the tongue area gain more illumination when compared to the face area. Based on this thresholding value the segmented tongue image is generated, and the results from this algorithm are shown in Figure 2.5.

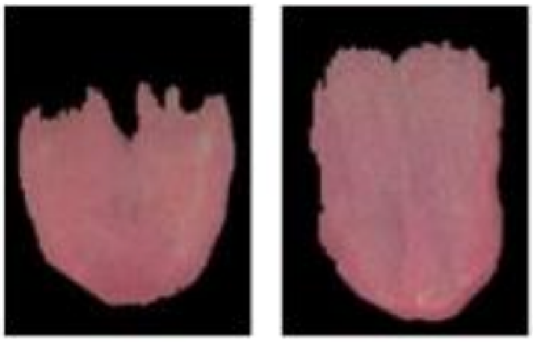


Figure 2.5   Flowchart of (Subandi *et al.*, 2019) segmentation algorithm

(Zhou *et al.*, 2019) proposed a tongue segmentation algorithm based on Deep Neural Networks. The segmentation algorithm is using the pixel information of the image for supervised deep CNN training. To begin, they introduced a feature pyramid network based on constructed context-aware residual blocks for the extraction of multi-scale tongue features. The extracted feature maps were then utilised to predict the ROIs of tongue candidates. Finally, ROI feature maps are used to achieve finer localisation and segmentation of the tongue body. The results from the aforementioned algorithm is shown in Figure 2.6.

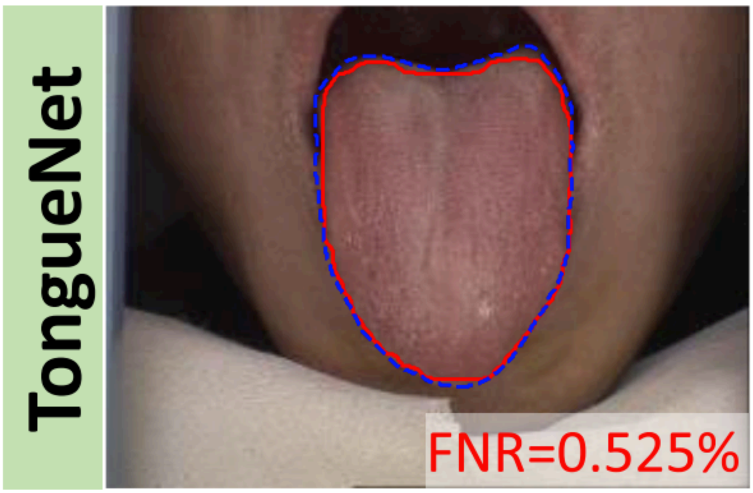


Figure 2.6   Flowchart of (Zhou *et al.*, 2019) segmentation algorithm

2.4 Convolutional Neural Networks based Segmentation

Convolutional Neural Networks (CNNs) are a subset of deep neural networks that are mostly used in the image processing area. It has the capability to classify and segment images with either supervised or unsupervised machine learning algorithms. CNN architecture consists of different layers. For classification application, CNN that consists of convolutional layers, max-pooling layers, and fully connected layers are used. Convolution and max-pooling layers are used for feature extraction. While convolution layers are intended for feature detection, max-pooling layers are designed for feature selection. Max-pooling layers are implemented for the applications which do not require high-resolution details. The output from convolution and pooling layers is the input for the fully connected layers for classification, which can be used for image classification, object detection, and facial recognition (Kumar, 2021).

CNN segmentation could be divided into semantic and instance segmentation. In the case of semantic segmentation, it classifies the image pixels into one or more classes which represents the objects in the image. In the case of the instance segmentation, each instance of objects in the image is identified with a specific ID. The difference is that instance does not associate each pixel with an object. CNN segmentation architecture employs encoder and decoder models. Encoders are used to convert the input into a representation that can be sent over the network, and decoders are used to reverse the process. With the goal of creating a segmentation map, encoders can be convolutional neural networks and decoders can be deconvolutional neural networks (Kumar, 2021). The flow of different layers of CNN in image segmentation is shown in Figure 2.7.

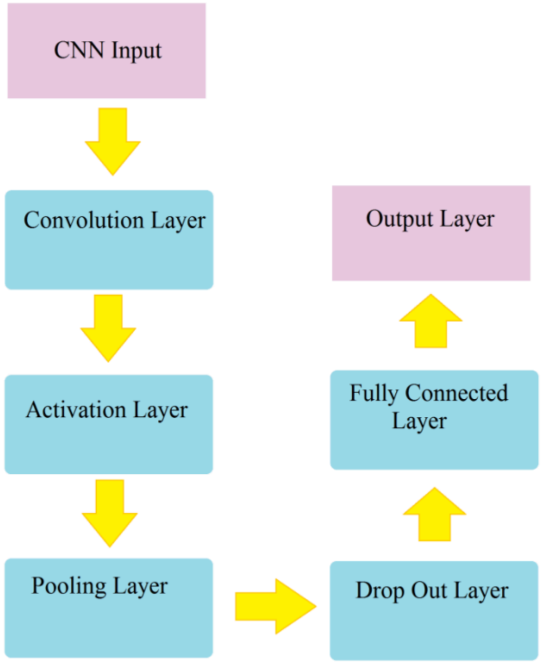


Figure 2.7 The flow of CNN layers in image segmentation (Kaushik and Kumar, 2019)

CNN-based segmentation algorithm has been used extensively on medical images. As shown by (Kadry *et al.*, 2022), they used various CNN methods to segment leukocytes from haematological images because conventional segmentation requires complex procedures to extract the required section of the image and the results obtained with conventional methods are sometimes poor depending on the type of the image. The results show that CNN-based segmentation can accurately segment leukocytes from haematological images. Generally, the segmentation is built by implementing a series of structured Convolutional Encoder-Decoder (CED) frameworks, with each framework transferring the learned information to the next section.

U-Net is a subset of CNN segmentation algorithm which is also used in medical images. A research by (Zhou *et al.*, 2019) shown an accurate and fast tongue segmentation algorithm based on U-Net layers. Similar to the CNN approach, U-Net consists of max-pooling and convolutional layers which are used to produce the segmented mask of the image. The unique part about U-Net architecture is that it consists of contraction and upscaling path as shown in Figure 2.8.

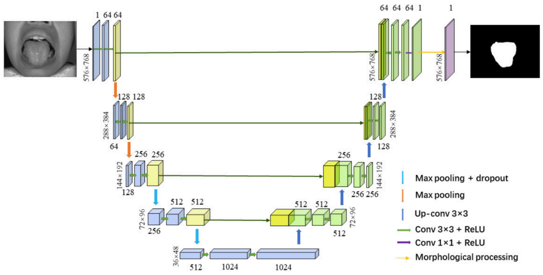


Figure 2.8 The U-Net architecture by (Zhou *et al.*, 2019)

(Futrega *et al.*, 2021) have proposed an optimised U-Net architecture that is able to segment brain tumour parts from MRI brain images. The model is created by searching for the optimal depth of the U-Net encoder, a number of convolutional channels and a post-processing strategy. The results of the algorithm are accurate as shown in Figure 2.9.

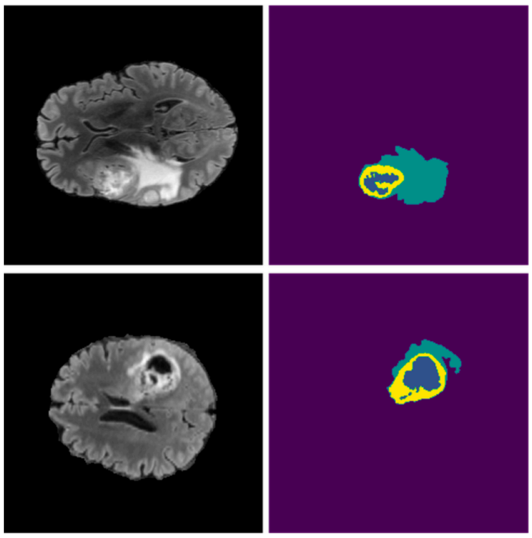


Figure 2.9 Segmentation results by (Zhou *et al.*, 2019)

Overall, it has been proven that CNN-based algorithms could be used for various medical images including tongue images for segmentation. There are many algorithms under CNN; however, U-Net architecture is shown to be the fastest and the most accurate.

2.5 Research Gap

There has been an increase in the research of automated healthcare systems recently and has seen a lot of advancement and development consequently. The research focuses on the development of better computerised tongue automated system; therefore, many researchers have proposed various technologies which improves the process of classifying the tongue. One of the main research was on utilising IoT and sensors involving more than one device for better results. However, the main limitation is that it does not include a friendly platform in which the users, including patients and doctors to use. Therefore, the project will help in improving the user experience by developing a mobile application integrated with the tongue diagnosis system.

Another area of research is the development of a tongue segmentation algorithm, that aims to remove the unrelated areas from the image and only leave the tongue for better diagnosis accuracy and allows for better analysis. One common limitation of the segmentation algorithm is that the taken image has to be captured from a specialised image acquisition device with a specific illumination setting to perform an accurate segmentation of the tongue; otherwise, it will produce poor segmentation results. Therefore, this project aims to develop an improved segmentation algorithm that works in various environments and produce an accurate results at the same time which also allows for the images captured by the smartphone to be segmented accurately.

Overall, the improved segmentation algorithm will enable the tongue diagnosis system to be used with any image acquisition device, including the camera on the mobile phones. Therefore, a mobile application will be created to utilise the improved segmentation algorithm, which will also improve the user experience of using this system.

2.6 Summary

It is a well-known fact among all researchers that healthcare is an absolute necessity for humans. There are many studies that work on improving the healthcare industry, especially with the latest discoveries and technologies. Tongue diagnosis is an aspect of TCM and general healthcare, researchers have been working on utilising the latest technologies like IoT and improving critical elements of the tongue diagnosis systems, which include the segmentation of the tongue. Recent research have shown that utilising IoT devices which are connected sensors to the system are necessary for the system to be more effective. The IoT concept states that the sensors collect the data, which are image sensors collecting the tongue images and processing the data. This will allow the data to be shared between various devices such as the mobile phone and Raspberry Pi.

Researchers have shown that the tongue segmentation algorithm is a necessary element of the computerised tongue diagnosis system along with the classification algorithm. Segmentation algorithms could be based on several techniques. (Wu and Zhang, 2015) proposed a segmentation algorithm based on combining the region and edge approaches to tongue segmentation. (Li *et al.*, 2019; Subandi *et al.*, 2019) has shown a segmentation algorithm that is based on colour decomposition and thresholding. Segmentation algorithms could be trained as deep learning models as shown by (Zhou *et al.*, 2019) research. However, the segmentation has a common limitation which is that the images have to be taken by a specialised image acquisition device. Therefore, this project's scope is to improve the segmentation algorithm. Moreover, it has been shown that deep learning methods could be used for other medical image segmentation as shown in (Kadry *et al.*, 2022; Futrega *et al.*, 2021) to segment leukocytes and the brain tumours in images, and U-Net architecture is a fast and accurate segmentation algorithm which has been illustrated by (Zhou *et al.*, 2019). Table 2.1 shows the summary of the segmentation algorithms that researchers have used to segment medical images.

Table 2.1 Summary of segmentation algorithm research and its features

Author	Algorithm	ROI	Algorithm description
(Wu and Zhang, 2015)	Edge detection	Tongue	Segmentation algorithm based on combining the region and edge approaches to tongue segmentation
(Li *et al.*, 2019)	Colour Thresholding	Tongue	Perform image thresholding on the transformed hue component int the colour space of the image
(Subandi *et al.*, 2019)	Colour Thresholding	Tongue	The threshold value is calculated based on the concept that the tongue area gains more illumination when compared to the face area
(Zhou *et al.*, 2019)	Deep Learning	Tongue	Pixels of the image are used for supervised deep CNN training
(Kadry *et al.*, 2022)	CNN	Leukocytes	The segmentation is built by implementing a series of structured Convolutional Encoder-Decoder frameworks, with each framework transferring the learned information to the next section
(Zhou *et al.*, 2019)	U-Net	Tongue	U-Net consists of max-pooling and convolutional layers which is used to produce the segmented mask of the image
(Futrega *et al.*, 2021)	U-Net	Brain tumour	The model is created by searching for the optimal depth of the U-Net encoder, the number of convolutional channels and the post-processing strategy

CHAPTER 3

METHODOLOGY

3.1 Project Workflow

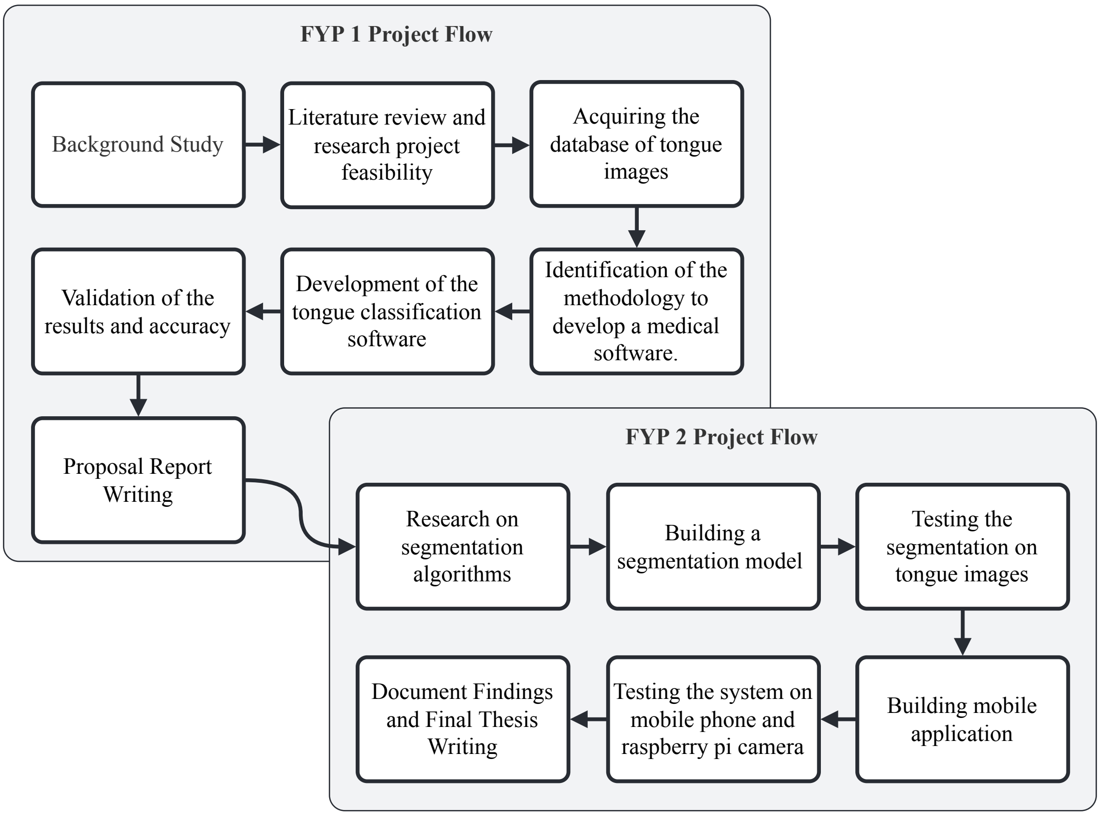


Figure 3.1 The workflow of the project

The project's workflow is depicted in Figure 3.1. The project starts with identifying the overview and objectives, then moves on to a literature review to determine the project's feasibility, then acquires the image data set needed to build the project. This is followed by identifying the methodology to achieve the project's objectives, and then building an application software to test the system with, and finally validating the system accuracy and writing the proposal report. Afterward, an extension to the project's scope is researched, which will use the Raspberry Pi and a mobile phone camera in conjunction with the algorithms developed in the previous stage. The segmentation algorithm is then improved to suit the mobile phone camera that will capture the image in the open space. The final thesis is then documented and presented.

3.2 Images Data Set

The images were taken by Tongue Image Analysing System (TIAS) on hundreds of outpatients at Japan's Oriental Medicine Research Centre, Kitasato University. TIAS is a closed-box acquisition system that includes several illuminators and a high-speed camera. There are 300 tongue images used in the segmentation, coating separation and classification procedures. Acquired tongue pictures were classified into three tongue colour types (light red, red, and deep red) according to a clinical diagnosis made by nine practitioners who have experience in TCM medicine of more than five years (Kamarudin, 2017).

3.3 Operation Flowchart

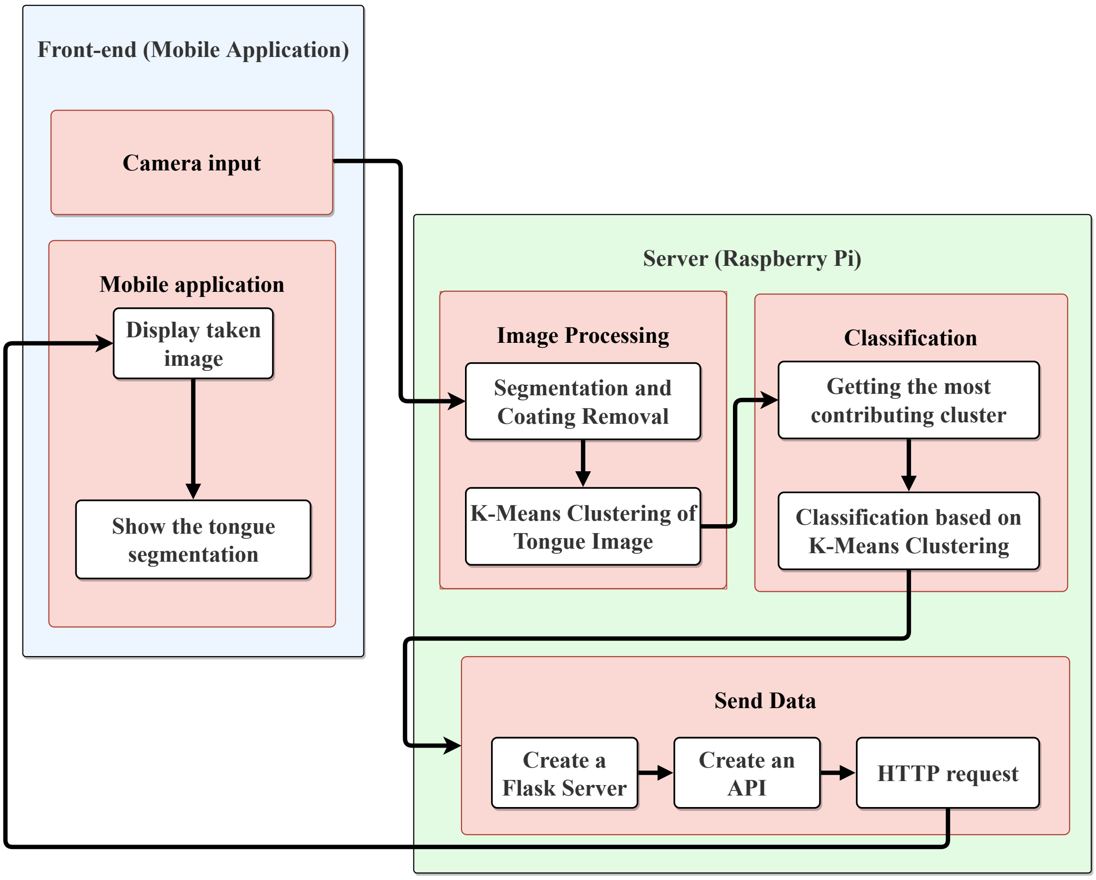


Figure 3.2 Overall flowchart of the developed system

As shown in Figure 3.2, the flow of the proposed system's operation starts with taking the input image, by a smartphone's camera through the mobile application. Then, the image is inputted to an image processing module where the segmentation algorithm of the tongue is executed. Following that is the classification algorithm which will be based on the K-Means and the most contributing colour in the tongue. Then, after the processing module is completed the results will be sent back to the mobile application to be displayed.

The classification software is written in C++. The software's flowchart is depicted in Figure 3.3. The process begins with taking the tongue image from the dataset as input, then the image is converted to HSV (Hue, Saturation, Value) colour space, which is required to perform the threshold-based segmentation and coating removal algorithms. Finally, the classification is performed using K-Means clustering algorithms, which is handled by obtaining the most contributing colour in the segmented image, which is then classified using threshold-based segmentation and coating removal algorithms.

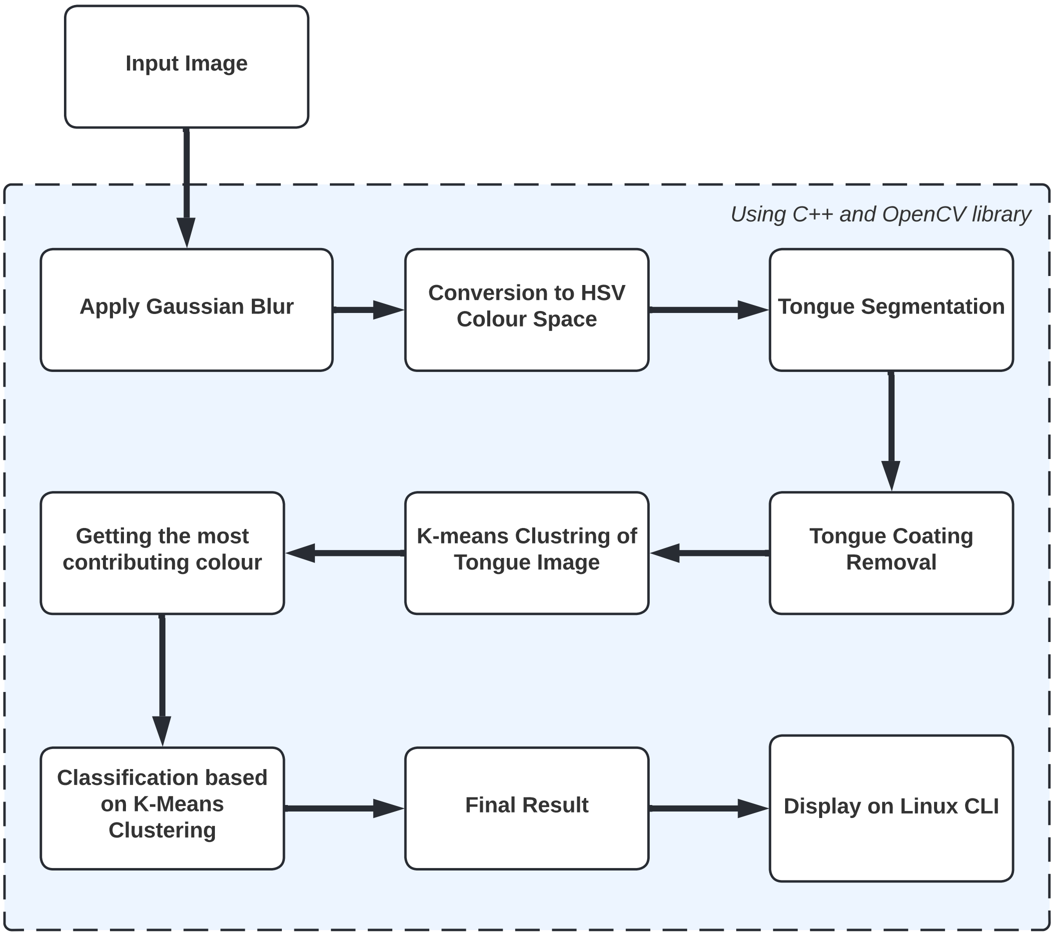


Figure 3.3 Flowchart of image cleaning and classification

3.4 Tongue Image Pre-processing

First, the input tongue image is blurred, which sharpens the edges of the edges for the segmentation process. The image is then converted to HSV colour space, which separates the light component (Value component) of the image and is used later in the segmentation process.

HSV colour space is created by performing a nonlinear transformation on the RGB colour space. As shown in Figure 3.4, RGB is described in a 3D Cartesian coordinate system, or a cube, where HSV is described in cylindrical coordinates, which results geometrically in a hexagonal cone. The HSV colour space is interpreted similarly to how humans see colours because it distinguishes between colour type (Hue and Saturation) and achromatic (Value) and represents them independently. This separation allows segmenting an image based on colour and lightness independently. As demonstrated by (Khediri *et al.*, 2021), in a comparative study between 5 major colour spaces, HSV performed better than other colour spaces, implying that HSV is the best option for segmentation.

The blurring and HSV conversion processes are performed on the HPS using the C++ OpenCV library, as shown in Figure 3.5. The blurring has sharpened the edges, and the HSV has highlighted the tongue area.

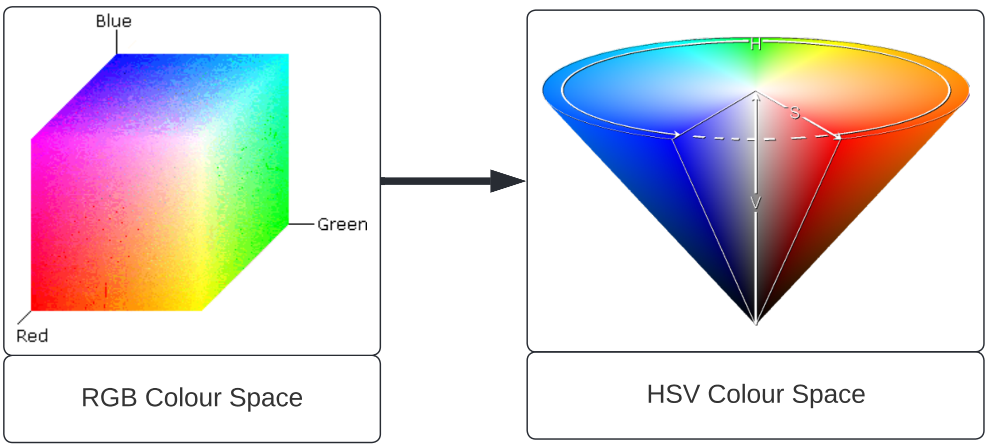


Figure 3.4 RGB and HSV Colour Spaces.

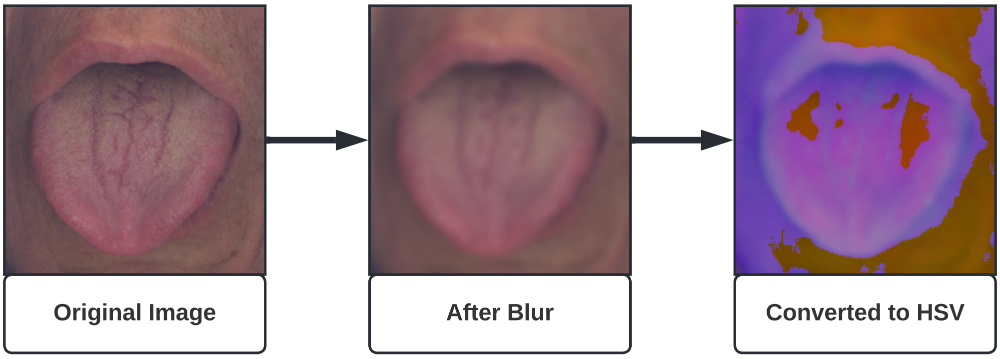


Figure 3.5 Conversion to HSV Colour Space.

3.5 Threshold-based Segmentation and Coating Removal

The image originally contains unwanted features such as the face and perioral areas surrounding the region of interest (the tongue area) which will be used in the classification algorithm; thus, a tongue segmentation algorithm is applied to the original image to extract the tongue area.

The segmentation process is shown in Figure 3.6, which starts with taking the V component (lightness) from the HSV colour space, since the segmentation is based on the Brightness Conformable Multiplier (BCM) proposed by (Kamarudin, 2017). BCM is derived to tolerate the inconsistent brightness of substance and perioral area due to the different posture of the tongue and illumination conditions of the image capturing system. This technique has the capability of eliminating the ineffective features on the tongue such as coloured coating, crackles, teeth marks and shadows that leads to false substance colour diagnosis. In short, BCM is a dynamic threshold value (ϵ) that is calculated based on the lightness of the image, this threshold value is calculated using (1), (2), and (3), where V is the V channel from HSV image and σ is the standard deviation of the image. After BCM is applied, Figure 3.6 shows the resulting image, in which the remaining parts of the perioral area are removed by selecting the largest area which is the tongue area.

$$V_{lower} = V_{mean} - \sigma \tag{3.1}$$

$$V_{upper} = V_{mean} + (\epsilon \times \sigma) \tag{3.2}$$

$$\epsilon = \begin{cases} V_{mean} & \text{if } V_{lower} < 2\sigma \\ V_{min} & \text{if } V_{lower} > 2\sigma \\ V_{mean}^2 & \text{otherwise} \end{cases} \tag{3.3}$$

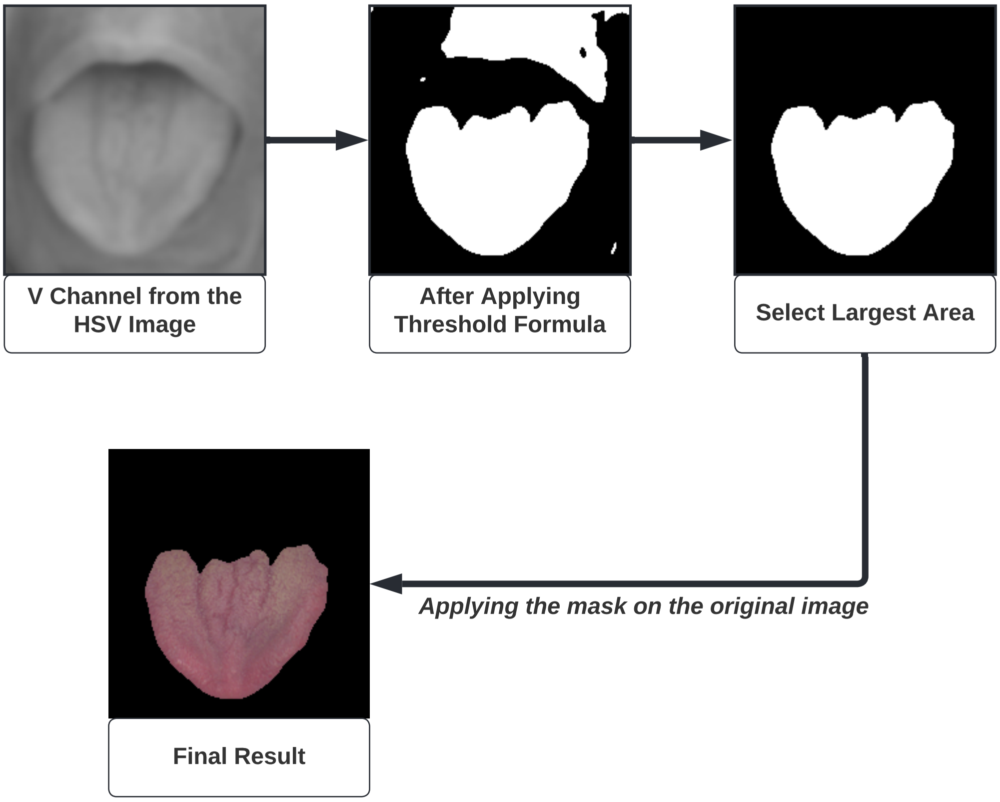


Figure 3.6 Segmentation Process.

The segmented tongue still has some coating areas that will affect the accuracy of the final diagnosis result; thus, a coating separation algorithm is used, as shown in Figure 3.7. The approach is based on the difference between colour parameters (H and S) and brightness parameter (V) of HSV colour space proposed by (Kamarudin, 2017) to improve the coating separation accuracy from tongue coating. The unrelated (coating) pixels are detected using (3.4). The segmentation and coating removal computations were done using thresholding functions found in the OpenCV library, which resulted in accurate segmented results.

$$coating_{pixels} = \frac{V - (S + H)}{3} \tag{3.4}$$

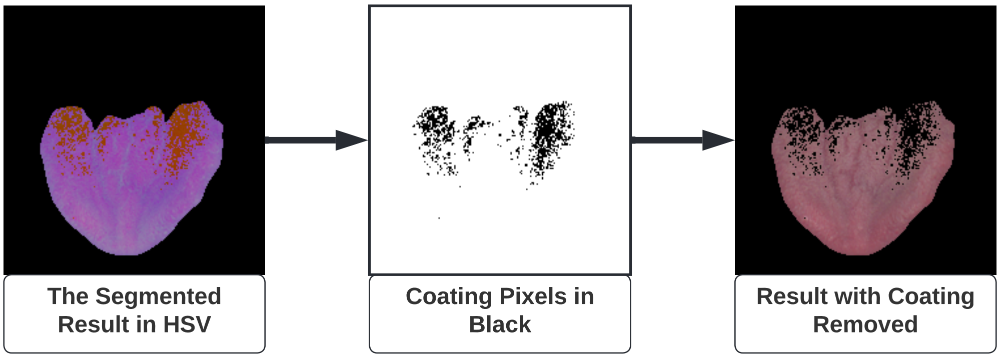


Figure 3.7 Coating Removal Process.

3.6 K-means Clustering and Classification

K-Means is a clustering algorithm which is used to divide a set of data into a predefined number of clusters, K clusters. This algorithm can be easily used for classification where the result clusters can be compared with a number of cases (Khan, 2017). The flow chart of the proposed K-means is shown in Figure 3.8.

It begins with capturing an image with the coating removed, as this is the object of interest for classification. The number of clusters chosen is four, with one for the background and three for the classification colours: red, light red, and deep red. The image colour space is converted to CIE L* a* b* as. The most contributing colour is then chosen and compared to a predefined Deep Red colour range for the final classification result.

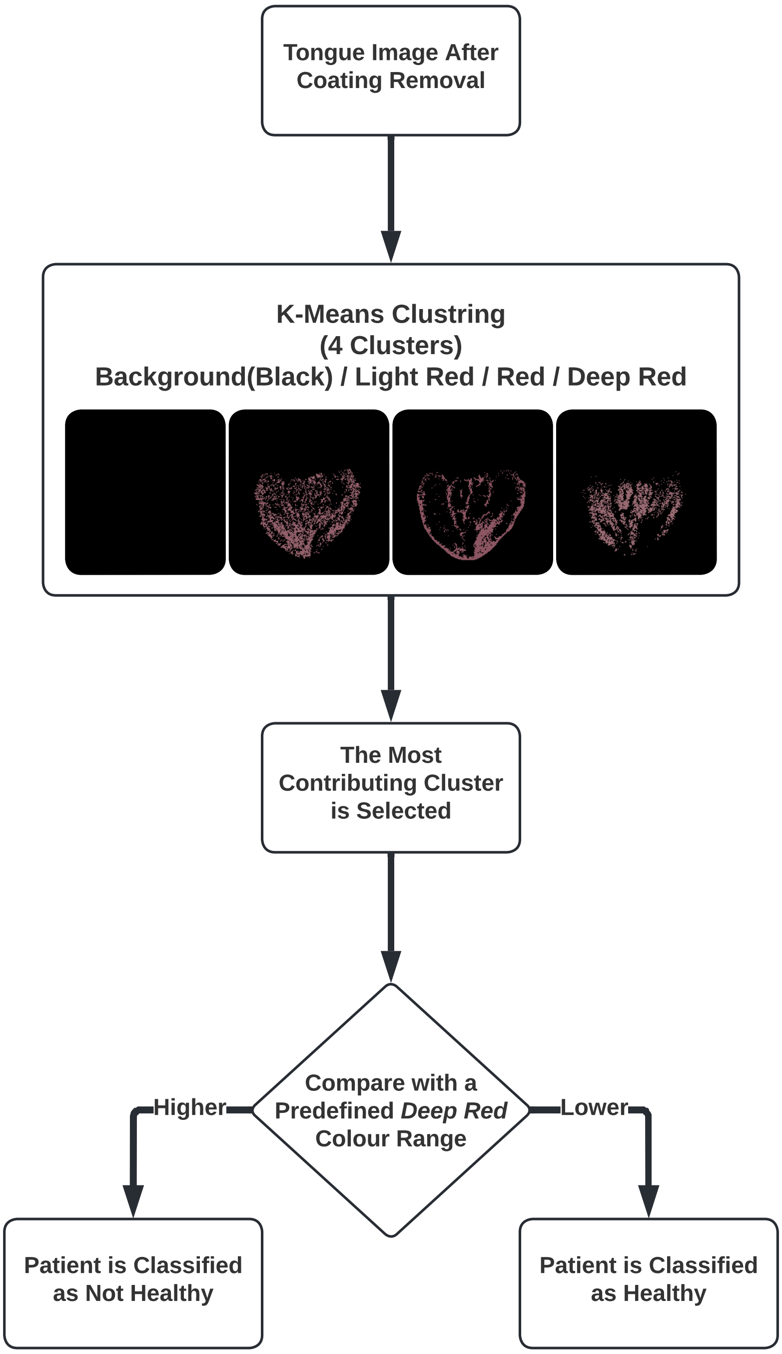


Figure 3.8 K-Means Classification Flow.

3.7 U-Net based Segmentation Algorithm

U-Net segmenation is based on the standard CNN architecture, the goal of the U-Net segmenation algorithm is to improve the scope where the tongue diagnosis system could be used. As in the threshold-based algorithm is limited to segmenting only tongue image taken from the specialised tongue image acquisition device; while the U-Net segmenation algorithm purpose is to extend the usage of the system to any image acquisition device like the smartphone's camera while keeping an accurate segmentation.

3.7.1 Data labelling

The U-Net segmentation algorithm is a supervised machine learning technique, which requires that the images has to be labelled first before training the model for semantic segmentation. The tongue images dataset has been labelled using a Python tool named "label-studio", as shown in Figure 3.9, the images is labelled manually using a brush tool which covers the tongue area only. This process is repeated on all the images in the dataset, which will be fed into the model training in the next stage.

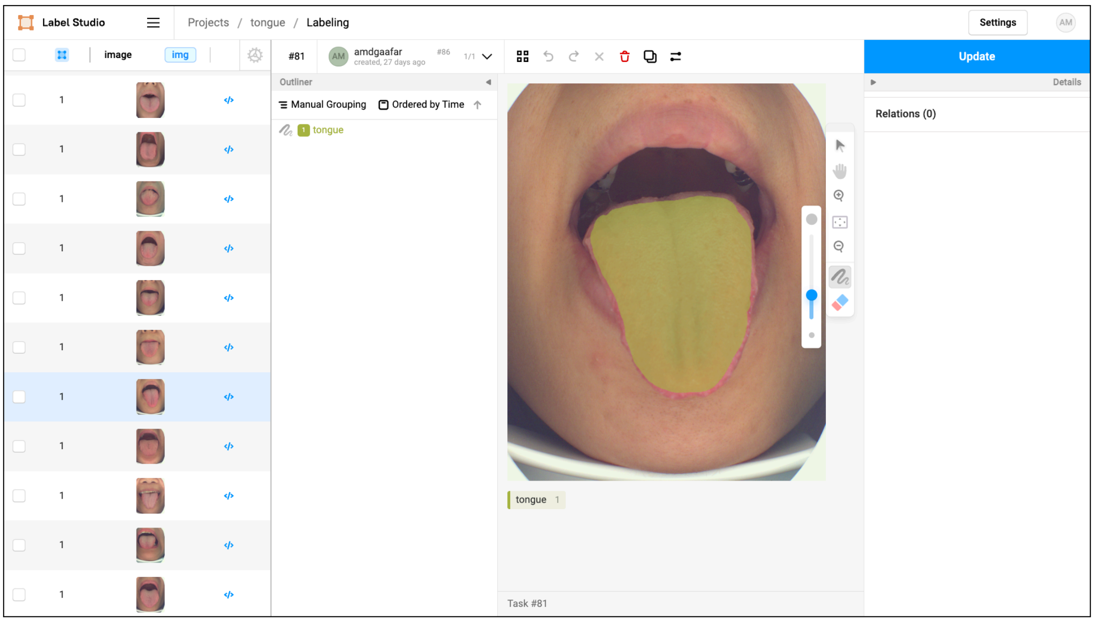


Figure 3.9 Labelling interface of Label-Studio software

3.7.2 Preparing U-Net Architecture for Training the Model

The U-Net is a CNN architecture that is able to perform image segmentation efficiently and at a fast speed. The U-Net algorithm was introduced by (Ronneberger *et al.*, 2015), which was designed for biomedical images. The algorithm consists of two paths, a contraction path to capture the context and a symmetric expanding path that allows for precise localisation.

The proposed U-Net network for this project is shown in Figure 3.10. The size of the input image is 128 × 128 pixels and the image has 3 channels. Then convolution operations are performed twice and then the resulting layer is gone through the maxpool operation with the size of 2×2, then the process is done in 4 iterations which is called contraction or the encoder path. And then in the expansion path, the up-convolutional operation is performed and the concatenation of feature maps helps to give the localisation information which is the segmented tongue picture.

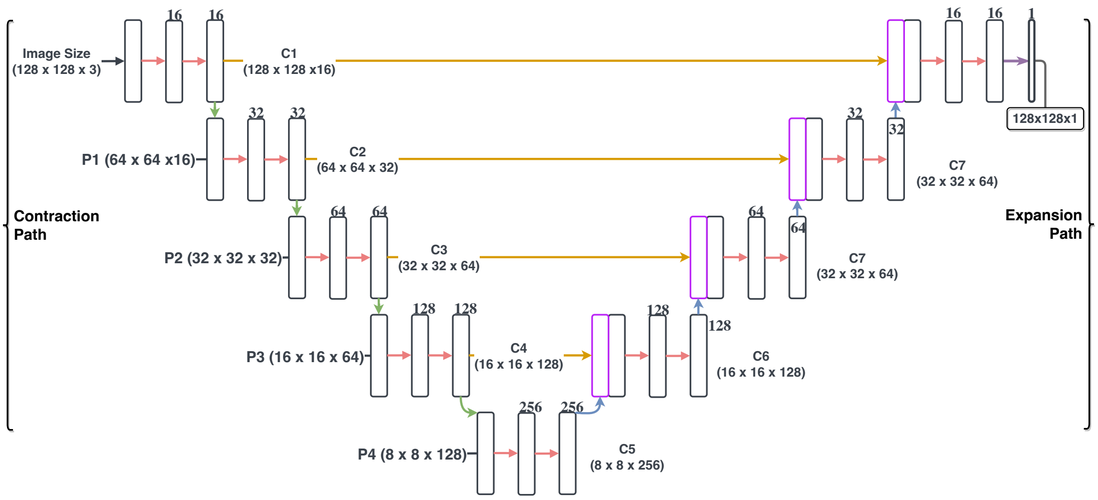


Figure 3.10 The U-Net architecture used for tongue segmentation

3.7.3 Training the Segmentation Model

After the tongue images dataset has been labelled and the U-Net segmentation model has been designed, the model is ready to be trained. The training is done using the TensorFlow python framework, which gives all the necessary functions needed to train a model (TensorFlow, 2022). The Keras library has been used in order to create the specific layers for the U-Net architecture, the activation function for the convolution, max-pooling, and up-convolutional layers have been set to Rectified Linear Unit (ReLU) activation function. The ReLU activation function is a non-linear function the will output the input if it is positive, otherwise, the output will be zero. This function is mostly used in neural networks, especially in CNNs and Multilayer perceptrons (iq.opengenus, 2021). The equation of the ReLU is Shown in Equation 3.5.

$$f(x) = max(o, x) \tag{3.5}$$

The output layer used the sigmoid activation function, which outputs a value between 0 and 1; inputs greater than 1 are transformed to the value 1, and inputs less than 0 are changed to the minimum value, 0 (Budaraju, 2020). The equation used to implement the sigmoid function is shown Equation 3.6.

$$f(x) = \frac{1}{1 + e^{(} - x)} \tag{3.6}$$

The reason for using the activation function is that they activate the neuron when the output reaches the function's set threshold value; they are a mathematical function that transforms the given input to the required output of a certain range. They are primarily in charge of controlling the state of the neuron. The neuron receives the sum of the product of inputs and randomly initialised weights, as well as a static bias for each layer. The activation function is applied to this sum to produce an output. Activation functions introduce nonlinearity, allowing the network to learn complex patterns in data such as images, text, videos, or sounds. Without an activation function, our model will behave similarly to a linear regression model with limited learning capacity (iq.opengenus, 2021). The python script used to generate the segmentation model is depicted in Figure 3.11.

```python
# Contraction path
c1 = tf.keras.layers.Conv2D(16, (3, 3), activation='relu',
    kernel_initializer='he_normal', padding='same')(s)
c1 = tf.keras.layers.Dropout(0.1)(c1)
c1 = tf.keras.layers.Conv2D(16, (3, 3), activation='relu',
    kernel_initializer='he_normal', padding='same')(c1)
p1 = tf.keras.layers.MaxPooling2D((2, 2))(c1)
...
...
...
# Expansive path
u6 = tf.keras.layers.Conv2DTranspose(128, (2, 2), strides=(2,
    2), padding='same')(c5)
u6 = tf.keras.layers.concatenate([u6, c4])
c6 = tf.keras.layers.Conv2D(128, (3, 3), activation='relu',
    kernel_initializer='he_normal', padding='same')(u6)
c6 = tf.keras.layers.Dropout(0.2)(c6)
c6 = tf.keras.layers.Conv2D(128, (3, 3), activation='relu',
    kernel_initializer='he_normal', padding='same')(c6)
...
...
...
outputs = tf.keras.layers.Conv2D(1, (1, 1), activation='
    sigmoid')(c9)
```

Figure 3.11 Python script for the designed U-Net architecture

3.7.4 Exporting the Model

After the training of the U-Net segmentation model is completed, the model need to be exported in order to be used in other devices. In this project, the segmentation model will be used in the Raspberry Pi. TensorFlow allows for the model to be saved for sharing purposes, with the option for saving the model manually. The function used to export the model is 'tf.keras.Model.save', which saves the entire model so that it can be used with the access of the original Python code. The model could be saved in two different formats, which are SavedModel and Hierarchical Data Format 5 (HDF5). To SavedModel is used with TensorFlow; however it could be also exported using HDF5 format which is a more general format (TensorFlow, 2023a).

HDF5 is an open-source file format that can handle large amounts of complex, heterogeneous data. HDF5 employs a "file directory"-like structure that allows to organise data within the file in a variety of structured ways, similar to how you might organise files on your computer. The HDF5 format also supports metadata embedding, making it self-describing (Leah A., 2017). Since the model is to be used in the Raspberry Pi, HDF5 format is chosen.

The command used to export the model is shown in Figure 3.12, which create a file with .h5 extension that contains all the trained model data. The model is then retrieved using the command shown in Figure 3.13, the argument of the function is the saved .h5 file exported from the previous command.

```
# Saves the entire model to a HDF5 file.
# The '.h5' extension indicates that the model is HDF5 format.
model.save('unet_tongue_segmentatino.h5')
```

Figure 3.12 TensorFlow command to export the model

```
# Recreates the exact same model
model = tf.keras.models.load_model('unet_tongue_segmentatino.
    h5')

# Show the model architecture
model.summary()
```

Figure 3.13 TensorFlow command to retrieve the model

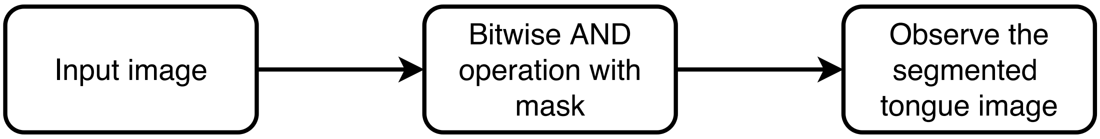


Figure 3.14 Segmentation evaluation process flow

3.8 Segmentation Accuracy Evaluation Process

The evaluation process is depicted in Figure 3.14 which relies on the concept of bitwise operation. The bitwise operator is used to perform bit by bit operations on binary patterns, mostly used in bit manipulation. These bitwise operators can be found in communication stacks, embedded software, low-level programming, search and optimisation problems (Techopedia, 2017).

In order to extract only the required part of the image, the bitwise operation is used, in which the pixels of value 0 will be removed from the image, and pixels of value 1 will remain in the image. The AND bitwise operation will perform a logical AND operation on each pixel on the image with the mask image, and the generated output will only be the ROI of the image.

OpenCV framework has built-in functions that can perform bitwise operations on images with a given mask, these functions are bitwise_and, bitwise_or, and bitwise_not. The format of the "bitwise_and" function is depicted in Figure 3.15, where src1 is the first image array, src2 is the second image array, dst is the output image array that has the same size and type as the input image, and mask is an optional operation mask, 8-bit single channel array, that specifies elements of the output array to be changed. The input tongue image is the inputted to the function along with the mask, which produces the new segmented tongue image, where its accuracy will be validated.

```
void cv::bitwise_and ( InputArray   src1,
                       InputArray   src2,
                       OutputArray  dst,
                       InputArray   mask = noArray()
                     )
Python:
   cv.bitwise_and( src1, src2[, dst[, mask]] ) -> dst
```

Figure 3.15 The bitwise_and function in OpenCV

3.9 Setting up the Raspberry Pi as Backend

3.9.1 Installing dependencies

The Raspberry Pi is first configured by installing the OpenCV dependencies on Linux. Since OpenCV is written in C++, a C++ compiler and a build tool are required. A tool to download OpenCV and unpack the source code is also required. In this case, 'wget' and 'unzip' are the tools for downloading and unpacking the source code respectively. And 'g++' and 'CMake' are the compiler and build helper respectively. The dependencies are downloaded and installed by running the command shown in Figure 3.16.

```
sudo apt install -y cmake g++ wget unzip
```

Figure 3.16 Install dependencies for OpenCV C++

The source code is downloaded from the OpenCV GitHub repository and unpacked since the downloaded source file is compressed. The OpenCV source code is downloaded and unpacked using the commands listed in Figure 3.17.

```
wget -O opencv.zip https://github.com/opencv/opencv/archive/4.
   x.zip
unzip opencv.zip
```

Figure 3.17 Download and unpack source code for OpenCV

The OpenCV binary must then be built from the source for the arm architecture in order to be installed on the Raspberry Pi, as the Raspberry Pi uses an arm processor. CMake is the tool used for building the binary. CMake is a helper tool that takes the source code and builds it using the C++ compiler for ARM. The commands shown in Figure 3.18 are used to build the binary for OpenCV.

```
# Create build directory
mkdir -p build && cd build

# Configure
cmake  ../opencv-4.x

# Build
cmake --build .
```

Figure 3.18 Building OpenCV from source

3.9.2 Creating Flask Server

Flask is a webframework that provides the tools, libraries and technologies that allows for creating APIs and web applications. It is a Python-based lightweight web application framework that is built on the WSGI toolkit and the Jinja2 template engine (Flask, 2022). In this project a Flask API is created to establish the communication between the Raspberry Pi and the mobile application using HTTP requests.

Two routes has been created; one is to send data and the other is to receive data which is the image captured from the mobile phone camera. The routes send and receive the images in base64 format. Base64 is a text-to-binary encoding scheme. It converts binary data into a printable ASCII string format by converting it to radix-64. It is mostly used to transmit binary data over media which does not handle the image formats correctly (SINGH, 2018). Therefore it is encoded to base64 before the image is sent through the http request and then after processing is decoded on both platforms the web and the mobile platform before it is displayed.

After the Raspberry Pi receives the image from the route, it performs the segmentation and the classification algorithms before it return the results back the mobile phone.

3.10 Tongue Diagnosis Mobile Application

There are many mobile application development frameworks availble, Flutter is the choosen framework to develop the mobile application for this project. Flutter is an open-source Software Development Kit (SDK) which enables for seamless cross-platform development, flutter is one of the best ways to create apps for both Android and iOS without having to write a separate codebase for each platform. The smartphone versions of these apps work as true native apps on Apple and Android devices and are compiled for each platform prior to publication. They do not require a browser or a runtime module. It is also possible to create web apps for browsers as well as native programmes for Windows, Linux, and macOS using the same codebase (Appify, 2021). In this project the development will be shown on Android platform. The role of the mobile application is to allow the tongue diagnosis to be performed in an open-space environment rather than limiting the computerised tongue diagnosis systems to be only used with specialised tongue image acquisition devices, and to show the capabilities of the segmentation algorithm in open-space settings.

The flow of using the mobile application is depicted in Figure 3.19. The first step of the mobile application is to capture the tongue image. This is achieved by utilising the image picker package that allows for opening the camera of the mobile phone and then use the captured image inside the mobile app. Then using http package, the image will be sent to the Raspberry Pi for processing the segmentation and classification algorithms, and then the mobile app receives the results to display.

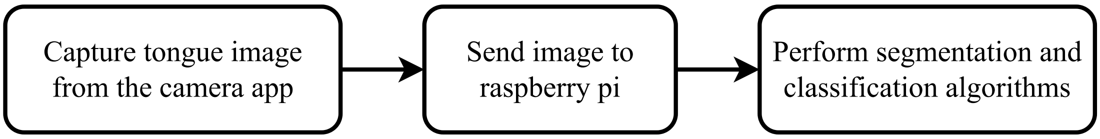


Figure 3.19 The flow of using the mobile application

3.11 Summary

In short, the system methodology starts with taking the input tongue image, the image will be captured from the phone camera through the developed mobile application. Then, the image will be processed for segmentation and classification. Two segmentation algorithms will be compared, the U-Net and threshold-based segmentation on images taken from the mobile phone to improve and determine the appropriate segmentation for images which are not taken from the specialised image acquisition device. Then, after the processing module is completed the results will be sent back to the mobile application to be displayed.

CHAPTER 4

RESULT AND DISCUSSION

The results of this project are the tongue image pre-processing or data cleaning and the classification results of the patient's condition, which are illustrated and discussed in the following sections. The results include a comparison between the threshold-based segmentation and U-Net based segmentation, the comparison will be based on how accurately each algorithm separates the tongue area from the background of the image. The images will be taken in various environments to investigate the segmentation capability of the U-Net segmentation algorithm.

4.1 Image Segmentation Results

The image segmenation algorithms implemented are the preprocessing part of any medical classification system, which separates the ROI from the background of the image to allow for greater analysis and produce more accurate results. This result section will illustrate the results of the Threshold based segmentation and the U-Net based segmentation for tongue images.

The ROI in this system is the tongue. Therefore, the designed system will separate the tongue element from the background of the image. This section will also illustrate the improvements resulting from the designed U-Net segmentation algorithm when compared with the threshold-based segmentation.

4.1.1 Threshold-based Segmentation

Figure 4.1 Loaded original image

In the case of when the image was taken from a specialised device, the illumination setting was set for the segmentation algorithm to be performed. The lighting is focused on the tongue as shown from the original image in Figure 4.1 which allows for the threshold value, BCM algorithm to segment only the tongue area.

Figure 4.2 Result of Segmentation and Coating removal algorithms

The original image has been segmented using the BCM algorithm, then the implemented coating removal algorithm is implemented coating removal shown in Figure 4.2.

However, when the algorithm is applied to images which are not taken from the specialised image acquisition device, the results are not accurate. Since the threshold-based segmentation relies mainly on the illumination system. Figure 4.3 shows a tongue image taken from a mobile phone (Samsung Galaxy Note 8), showing that the segmentation is not accurate.

When the tongue area is close to the camera, as shown in Figure 4.4 focusing only on the tongue, the resulting segmentation is accurate; however, this shows that the threshold-based algorithm is not stable when the images used are not taken from a specialised image acquisition device with sufficient lighting.

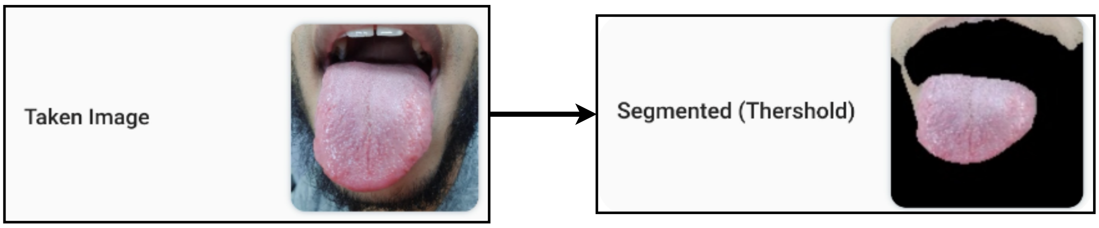


Figure 4.3 Threshold-based segmentation result when mobile phone camera is used

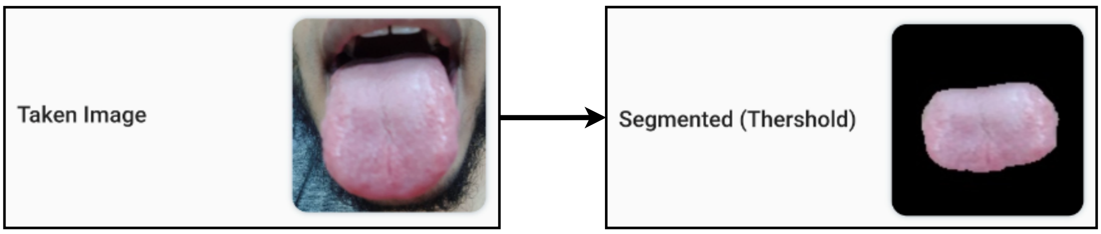


Figure 4.4 Threshold-based segmentation result with tongue close to the camera when mobile phone camera is used

One more limitation when using the threshold-based algorithm is when the background area is brighter than the tongue area, the algorithm removes the tongue area and keeps the area which has more brightness. As shown in Figure 4.5, the background of the image taken is bright, so the segmented area is the background which is the opposite of the desired results.

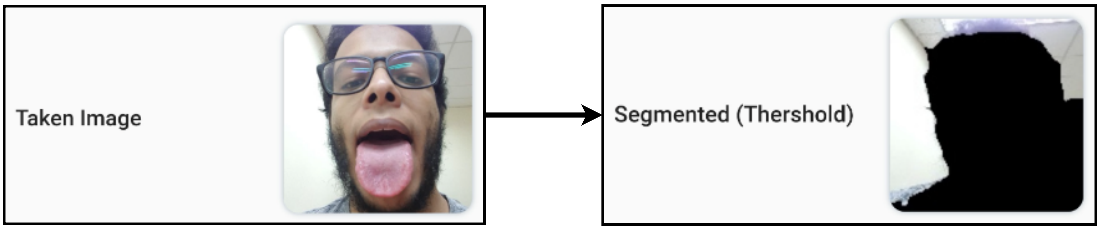


Figure 4.5 Threshold-based segmentation result with bright background when mobile phone camera is used

When the image taken does not contain any tongue element in the picture taken, the expected output should be a black image indicating that there is no tongue found in the image to be segmented. However, the threshold-based segmentation algorithm does segment few areas which have the same hue and light value as the tongue as shown in Figure 4.6.

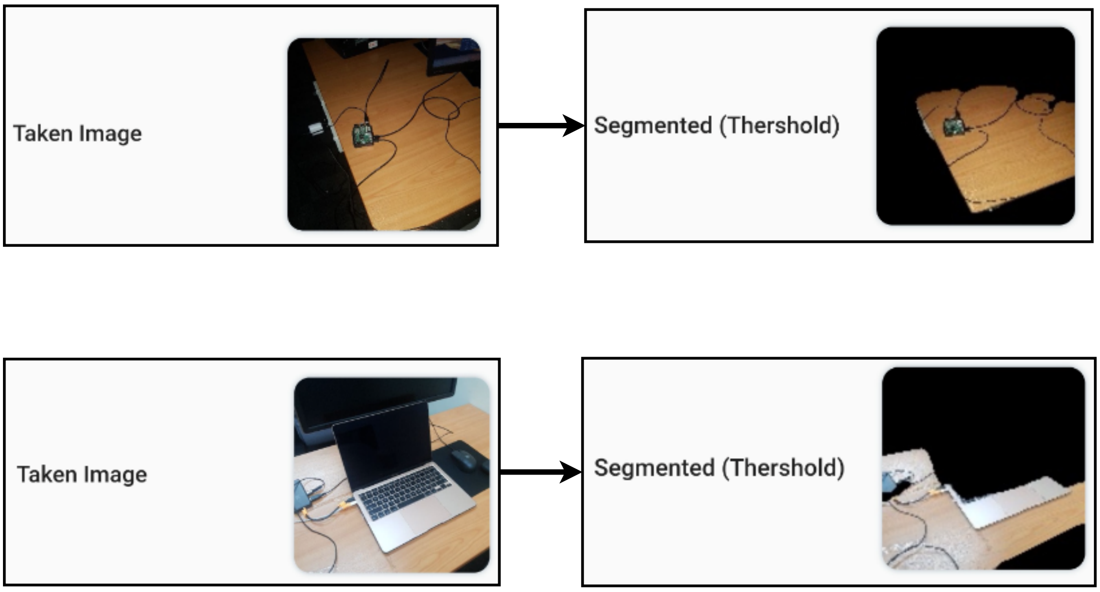


Figure 4.6 Threshold-based segmentation result when no tongue exists in the image

4.1.2 U-Net based Segmentation

To address the aforementioned limitations of threshold-based segmentation, a new CNN-based segmentation algorithm named U-Net has been developed and trained on the dataset, and the results are discussed in this section.

The model has been trained using the data dataset that has been labelled, a total of 357 images has labelled been used for training and testing the model. For training the model, 342 images has been used as a training dataset, the training dataset is divided into a training and validation data set. The validation is not used for training; however, it is used for validating the accuracy while training without exposing the output to the model and is needed to provide an indication when the training will stop. The training is 90% and the validation is 10% of 342 images. And the remaining of the dataset is 15 tongue images which have not been exposed to the model during the training which is used for testing the model performance and ensuring it is accurate after the training is finished.

After the model has been trained, the model is tested on the training, validation, and testing dataset. The output of the model is tabulated in Table 4.1, which shows the images count that has been accurately segmented and how many are not segmented accurately. In summary, the accuracy when the images within the dataset is used is 99.41% and for the testing dataset, the accuracy is 93.33%.

Table 4.1 The results of training the model

<table>
<tr><td colspan="7">From images which has been taken from a specialised tongue image acquisition device
U-Net</td></tr>
<tr><td rowspan="2">Image type</td><td colspan="2">Prediction result</td><td colspan="4"></td></tr>
<tr><td>Accurate</td><td>Not accurate</td><td>Total images</td><td>Accuracy</td><td rowspan="3">Average accuracy</td><td rowspan="3">96.37%</td></tr>
<tr><td>Within the dataset</td><td>340</td><td>2</td><td>342</td><td>99.42%</td></tr>
<tr><td>Testing</td><td>14</td><td>1</td><td>15</td><td>93.33%</td></tr>
<tr><td>Total</td><td>354</td><td>3</td><td>357</td><td colspan="3"></td></tr>
<tr><td colspan="7">This indicate that the model has been successfully trained based on the dataset</td></tr>
</table>

Figure 4.7 shows a sample of the output after training when an image is taken from a specialised image acquisition device, the original input image is shown with the output segmentation. And for comparison, the manual label that has been used for training and the prediction output mask is shown, which indicates that the accuracy is very high. The final segmented output is generated by combining the original image and the prediction mask for better observation of the output.

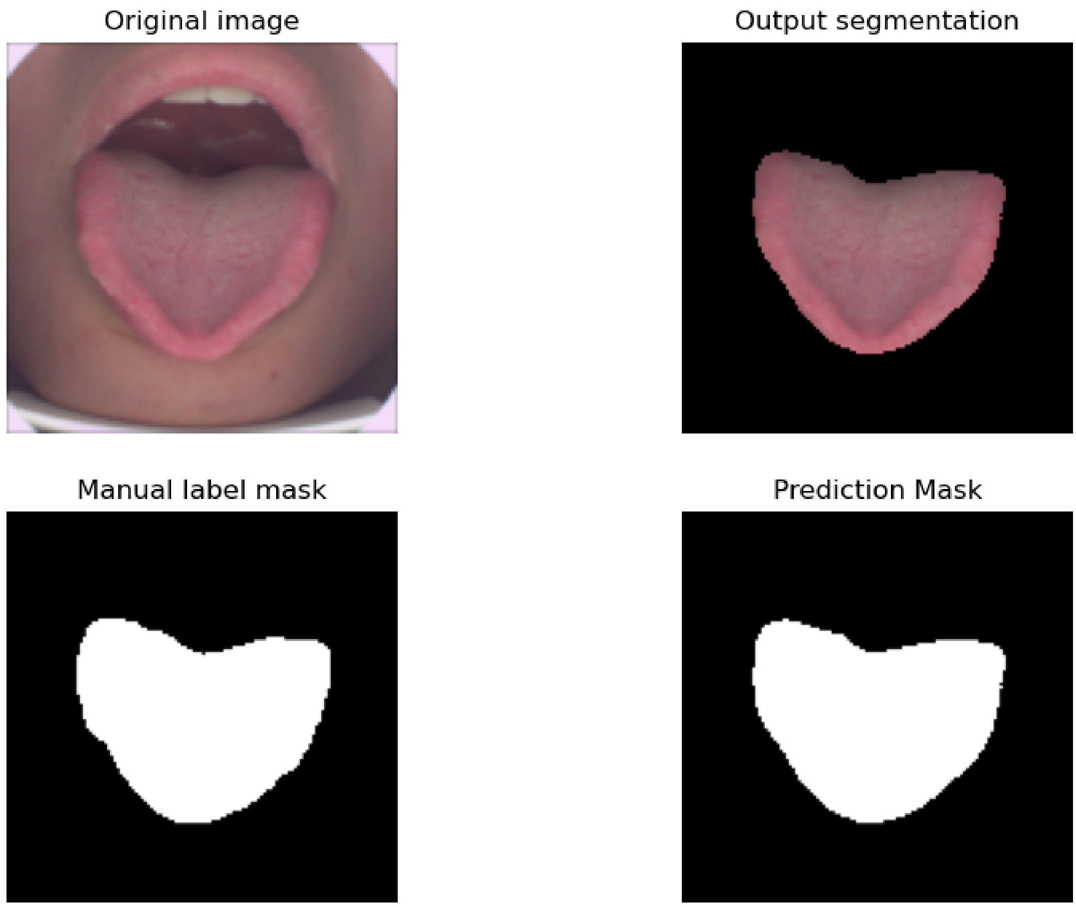
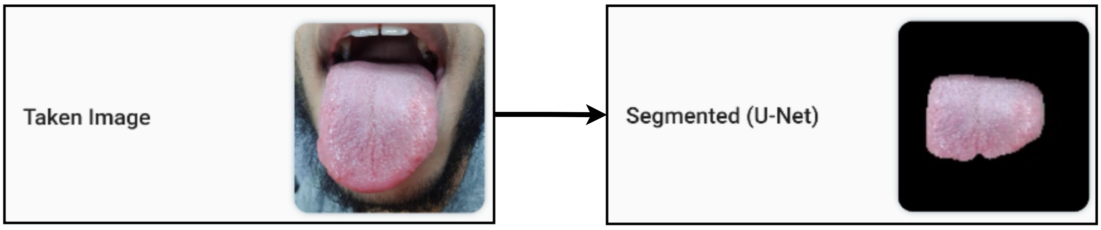
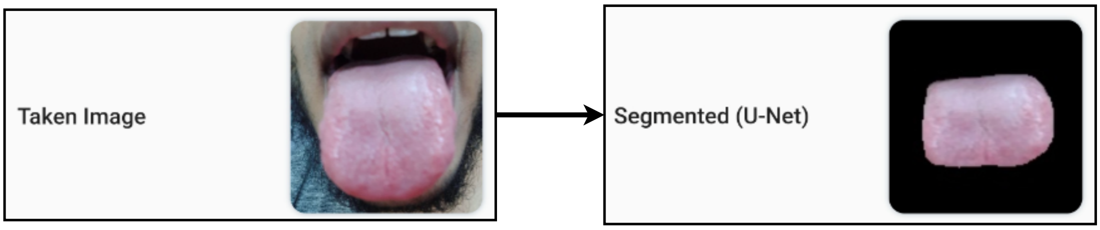


Figure 4.7 A sample of the output segmentation and the prediction mask after training the U-Net model

The input image is taken from the mobile phone camera through the developed mobile phone appliances. In the case of when the tongue is close to the camera making the tongue take the largest area of the taken image the segmentation result is accurate as shown in Figure 4.8 where the segmented image is only the tongue image.



(a) First image result for U-Net segmentation when the image taken is near the camera



(b) Second image result for U-Net segmentation when the image taken is near the camera

Figure 4.8 U-Net segmentation result when the image taken is near to the camera

When the taken image is taken far from the phone camera, it will make the tongue area in the image smaller and it will add more background noise in the picture to segment. However, the U-Net segmentation manages to segment the tongue area precisely in such situations as shown in Figure 4.9. Furthermore, the tongue image taken has a bright background, which makes threshold-based segmentation difficult to separate the tongue area; however, U-Net-based segmentation detects the tongue not only based on brightness but also texture and size, making the segmentation successful.

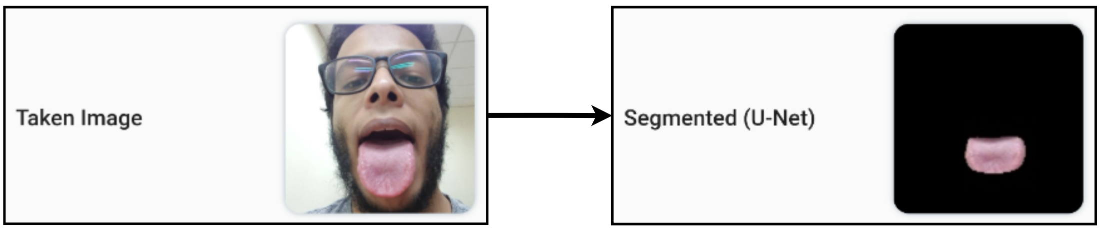
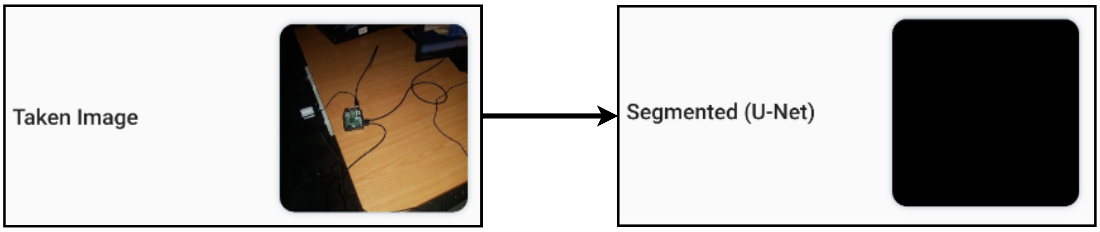
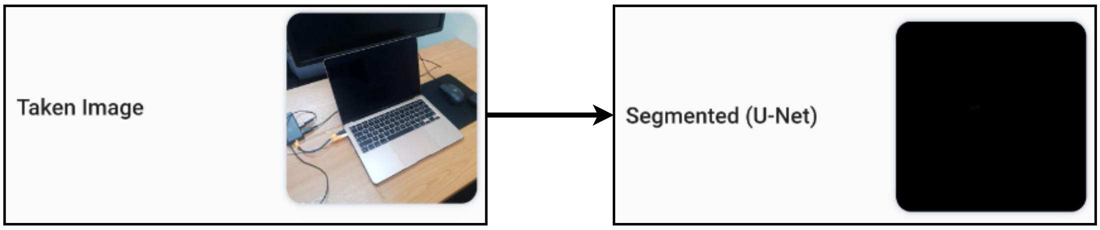


Figure 4.9 U-Net segmentation result when the image taken is far from the camera

In the event that the taken tongue image contains no tongue, the U-Net segmentation is able to detect that there is no tongue detected in the image and therefore the resulting the image to be black as shown in Figure 4.10 which illustrated two images without tongue along with the results. That is because the U-Net is able to identify the tongue based on its shape and size rather than only the colour, as opposed to the threshold-based segmentation which attempts to segment based on hue and brightness colour values, which can produce misleading results.



(a) First image result for U-Net segmentation when the image taken contains no tongue



(b) Second image result for U-Net segmentation when the image taken contains no tongue

Figure 4.10 U-Net segmentation result when the image taken contains no tongue

4.1.3 Comparison and Discussion

When comparing the two algorithms, it can be observed that the U-Net based segmentation adds a lot of improvement to the final segmented tongue image in various environments in open space, which eliminates the need for a specialised image acquisition device. The two algorithms have been tested on 60 images taken from a phone camera and the results are tabulated in Table 4.2 and 4.3.

Table 4.2 illustrates the results of the threshold-based segmentation algorithm which indicates that when this algorithm is used in an environmental condition other than when the image is taken from the specialised image acquisition device, it becomes unusable and produces wrong prediction in most of the images resulting in an average accuracy of 3.75%. This is because the phone camera introduces many colours and lighting condition which the algorithm does not count.

Table 4.2 Threshold-based segmentation results with the phone camera images

Images which has been taken from the phone camera through the mobile app Threshold				
Condition	Segmentation			
	Accurate	Not accurate	Total	Accuracy (%)
Near	3	17	20	15.00%
Far	0	15	15	0.00%
Bright	0	15	15	0.00%
No tongue	0	10	10	0.00%
Total	3	57	60	Average accuracy (%)
				3.75%

On the same images, the U-Net algorithm is executed and has shown an accuracy improvement from 3.75% to 76.67% as shown in the results in Table 4.3. The U-Net segmentation is trained to identify the structure of the tongue rather than only the colour and the lighting condition, which is the reason for this accuracy improvement. This algorithm is more suitable to be used in open space environments when compared to the threshold-based segmentation.

Table 4.3 U-Net segmentation results with the phone camera images

Images which has been taken from the phone camera through the mobile app U-Net				
Condition	Segmentation			
	Accurate	Not accurate	Total	Accuracy (%)
Near	20	0	20	100.00%
Far	8	7	15	53.33%
Bright	11	4	15	73.33%
No tongue	8	2	10	80.00%
Total	47	13	60	Average accuracy (%)
				76.67%

4.2 Clustering-based Classification Algorithm

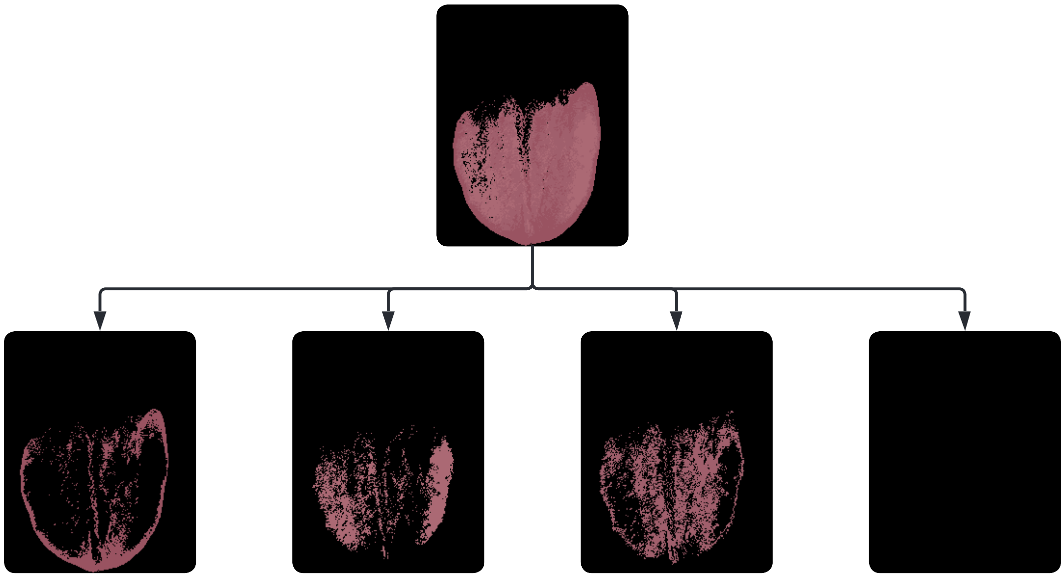


Figure 4.11 Results of K-Means clustering algorithm

The K-Means is applied with $K = 4$ on the image that has been cleaned from unwanted features, such as coating. The results are illustrated in Figure 4.11, showing

Table 4.4 Colour range used for classification

		Colour range for each channel in CIE L*a*b* colour space		
		L	a	b
Image label	Deep Red	$146 \geq L \geq 114$	$157 \geq a \geq 145$	$136 \geq b \geq 130$
	Red	$163 \geq L \geq 68$	$165 \geq a \geq 137$	$140 \geq L \geq 121$
	Light Red	$159 \geq L \geq 67$	$167 \geq a \geq 138$	$141 \geq b \geq 120$

4 clusters as expected. The cluster with the most contributing colour is then used for classification which is discussed in Section 4.2.1.

4.2.1 Classification Results

The algorithm is executed on all the labelled tongue images from the database in order to get the most contributing colour for each image, which resulted in the following colour ranges shown in Table 4.4 to be used in the classification.

Then, the results of these colour ranges are used to classify the tongue, the healthy tongue is associated with deep red colour ranges, and the healthy tongue is associated with a red and a light red colour ranges. The implemented classifier resulted in 76.47% accuracy when classifying healthy patients and 73.44% when classifying unhealthy patients.

The confusion matrix in Table 4.5 shows a more detailed output on the classifier implemented, out of 241 healthy tongue images, 177 images were classified correctly while classifying healthy tongue images. And out of 34 unhealthy tongue images, 26 were classified correctly.

Table 4.5 Confusion Matrix showing the accuracy of the classifier

		Predicted		Total
		Healthy	Not healthy	
Actual	Healthy	177	8	185
	Not healthy	64	26	90
Total		241	34	

4.3 System Efficiency Evaluation

The performance of the Raspberry Pi as a server is evaluated based on the execution time it takes to finish the segmentation. A comparison between the U-Net based segmentation and the threshold-based segmentation will be illustrated and discussed. The execution time is calculated using a python function which calculates the time at the beginning of the execution and the end of the execution and then calculates the difference resulting in the execution time of the algorithm. The python script is depicted in Figure 4.12.

```python
# Unet Pred
start_time = time.time() # timer stars
pred_mask = model.predict(nparr)
pred_mask_t = (pred_mask > 0.5).astype(np.uint8)
segUnetImg = cv2.bitwise_and(img, img, mask = pred_mask_t[0])
end_time = time.time() # timer stops
print(f'Unet: {end_time - start_time}')
```

Figure 4.12 Python script for the calculating the execution time

4.3.1 Segmentation Execution Time Comparison

The execution time of each algorithm is illustrated in Table 4.6. Total of 10 tongue images has been captured and the execution time is calculated on the two algorithms for the same images on the same platform. The average execution time for the U-Net segmentation algorithm is 1.2408 *seconds* and for the threshold-based segmentation is 0.4378 *seconds*, which shows that using the U-Net segmentation comes only with one compromise, which is the longer execution time. This might limit the applications of this algorithm to be used in real-time.

Table 4.6 The execution time of U-Net and threshold-based segmentation algorithm

Scan No.	U-Net (sec)	Threshold (sec)
1	1.9175	0.6425
2	1.2044	0.4637
3	1.3071	0.3870
4	1.1167	0.4366
5	1.0986	0.4119
6	1.1554	0.4183
7	1.1296	0.3974
8	1.2138	0.3820
9	1.1381	0.4081
10	1.1267	0.4308
Average	1.2408	0.4378

4.4 Summary

In summary, the U-Net segmentation algorithm's results show that it can be used in a variety of open environments that allow for the use of tongue diagnosis systems on mobile devices since the mobile phone camera does not always have the illumination settings required for threshold-based segmentation algorithms. The environment conditions tested where the image is taken are when the tongue is close to and far from the camera, images with a very bright background, and whether or not the algorithm can detect if there is a tongue in the image taken. By using the U-Net segmentation, the accuracy is improved from 3.75% to 76.67% when the mobile phone camera is used.

CHAPTER 5

CONCLUSION

5.1 Outcomes

In conclusion, the tongue diagnosis system has been developed, which includes a cleaning algorithm for tongue images which is the improved U-Net based segmentation algorithm to remove the unwanted features from the image, and a classification algorithm that classifies the condition of a patient from a tongue image. An unhealthy classification indicate that the patient may few health issues such as blood congestion, water imbalance, or psychological problems, while a healthy classification indicates otherwise. Based on the conducted literature review, segmentation is a key component in any computerised automated medical classification system, as it will remove unrelated elements to the diagnosis which will aid in improving the classification accuracy.

The proposed U-Net based segmentation algorithm managed to segment the tongue element in various environments accurately, which eliminates the need for specialised image acquisition devices. The developed mobile application has shown that the user is able to take images in various locations and keep an accurate segmentation. The trained segmentation model showed an average accuracy of 96.37% from images which were taken from the specialised image acquisition device, and an accuracy improvement from 3.75% to 76.67% for images taken from the smartphone in various environments. The threshold-based segmentation and the U-Net segmentation model have been compared; the threshold-based segmentation is not usable with images taken from the smartphone as it introduces new colour ranges which are not counted in the threshold, whereas the U-Net segmentation is able to segment the images taken from the smartphone's camera allowing for better user experience and larger application scope. The classification utilised the K-means algorithm, achieving an accuracy of 76% for healthy patients and 73% for unhealthy patients.

5.2 Future Works

5.2.1 CNN Model Accuracy

As for the future works, generally, the segmentation model could be improved by adding more tongue images in the trained dataset, so that the model produces stronger and more accurate in more situations. To accomplish this, adding an annotation tool on the developed mobile application to allow correcting miss segmented areas of the tongue and then re-train the model on the mobile phone using TensorFlow lite, which is a python library used mainly on mobile phones considering the limited processing power of the mobile phone (TensorFlow, 2023b).

Another method to improve the segmentation is to detect the tongue object boundary in the image and then zoom into the detected boundary before performing the segmentation algorithm. This will generate a new image with less background noise around the tongue, then after the image has been generated, the segmentation algorithm will begin processing to segment the tongue. This method could be developed using YOLO object detection algorithm, which has the capability to draw boundaries on the image (Neelam, 2021).

5.2.2 Execution Time Acceleration

The processing time for the U-Net segmentation model could be improved, this could be done by utilising FPGA processing and parallelism power which can be useful when the model prediction command is executed. Researchers proved that utilising GPU or FPGA could accelerate the computation of CNN algorithms (Wang *et al.*, 2022).

5.2.3 Expand Classification Scope

TCM is used to classify many diseases, generally a tongue with deep red colour range signifies that the patient in not healthy and should seek medical diagnosis from professionals for more details on the specific diseases of the patient. However, the scope of classification could be extended based on other factors like the tongue size, texture, coating size, etc. These information could help in identifying specific diseases like cancer as shown by (Lo *et al.*, 2015), where they showed that automated tongue diagnosis could be used to detect early-stage breast cancer.

REFERENCES

Anastasi, J. K., Currie, L. M. and Kim, G. H. (2019). Understanding Diagnostic Reasoning in TCM Practice: Tongue Diagnosis. *Alternative Therapies*. 15(3), 11.

Appify, D. (2021). *Flutter App Development - Why You Should Choose Flutter*. Retrievable at `https://www.appify.digital/post/flutter-app-development`.

Budaraju, D. (2020). *Activation Function Sigmoid*. Retrievable at `https://medium.com/analytics-vidhya/activation-function-sigmoid-7673dc0efcbe`.

Flask (2022). *Flask Documentation (2.2.x)*. Retrievable at `https://flask.palletsprojects.com/en/2.2.x/`.

Futrega, M., Milesi, A., Marcinkiewicz, M. and Ribalta, P. (2021). *Optimized U-net for brain tumor segmentation*.

Haghi Kashani, M., Madanipour, M., Nikravan, M., Asghari, P. and Mahdipour, E. (2021). A systematic review of IoT in healthcare: Applications, techniques, and trends. *Journal of Network and Computer Applications*. 192, 103164. ISSN 1084-8045. doi:10.1016/j.jnca.2021.103164. Retrievable at `https://www.sciencedirect.com/science/article/pii/S1084804521001764`.

iq.opengenus (2021). *ReLU (Rectified Linear Unit) Activation Function*. Retrievable at `https://iq.opengenus.org/relu-activation/`.

Ismail, Y. (2019). *Internet of Things (IoT) for Automated and Smart Applications*. Rijeka: IntechOpen. ISBN 978-1-78984-096-4. doi:10.5772/intechopen.77404. Retrievable at `https://doi.org/10.5772/intechopen.77404`.

Jung, C. J., Jeon, Y. J., Kim, J. Y. and Kim, K. H. (2012). Review on the current trends in tongue diagnosis systems. *Integrative Medicine Research*. 1(1), 13–20. ISSN 2213-4220. doi:10.1016/j.imr.2012.09.001. Retrievable at `https://www.sciencedirect.com/science/article/pii/S2213422012000029`.

Kadry, S., Rajinikanth, V., Taniar, D., Damaeviius, R. and Valencia, X. P. B. (2022). Automated segmentation of leukocyte from hematological imagesa study using various CNN schemes. *The Journal of Supercomputing*. 78(5), 6974–

6994. ISSN 1573-0484. doi:10.1007/s11227-021-04125-4. Retrievable at https://doi.org/10.1007/s11227-021-04125-4.

Kamarudin, N. D. B. (2017). *Tongue Disease Diagnosis based on Brightness Conformable Multiplier thresholding and K-means Support Vector Machine classifier*. Master's Thesis. Universiti Teknologi Malaysia.

Kaushik, R. and Kumar, S. (2019). Image Segmentation Using Convolutional Neural Network. *TERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH*. 8(11).

Khan, M. (2017). *KMeans Clustering for Classification*. Retrievable at https://towardsdatascience.com/kmeans-clustering-for-classification-74b992405d0a.

Khediri, N., Ammar, M. B. and Kherallah, M. (2021). Comparison of Image Segmentation using Different Color Spaces. In *2021 IEEE 21st International Conference on Communication Technology (ICCT)*. October. 1188–1192. doi: 10.1109/ICCT52962.2021.9658094. ISSN: 2576-7828.

Kirschbaum, B. (2000). *Atlas of Chinese tongue diagnosis*. Seattle: Eastland Press. ISBN 978-0-939616-33-6.

Kumar, A. (2021). *CNN Basic Architecture for Classification & Segmentation*. Retrievable at https://vitalflux.com/cnn-basic-architecture-for-classification-segmentation/.

Leah A., W. (2017). *Hierarchical data formats*. Retrievable at https://nceas.github.io/oss-lessons/spatial-data-gis-law/5-tues-intro-hierarchical-formats.html.

Lee, M., Lee, H., Kim, Y., Kim, J., Cho, M., Jang, J. and Jang, H. (2018). Mobile App-Based Health Promotion Programs: A Systematic Review of the Literature. *International Journal of Environmental Research and Public Health*. 15(12), 2838. ISSN 1660-4601. doi:10.3390/ijerph15122838. Retrievable at https://www.mdpi.com/1660-4601/15/12/2838, number: 12 Publisher: Multidisciplinary Digital Publishing Institute.

Li, Z., Yu, Z., Liu, W., Xu, Y., Zhang, D. and Cheng, Y. (2019). Tongue image segmentation via color decomposition and thresholding. *Concurrency and*

Computation: Practice and Experience. 31(23). ISSN 1532-0626, 1532-0634. doi:10.1002/cpe.4662. Retrievable at https://onlinelibrary.wiley.com/doi/10.1002/cpe.4662.

Lo, L.-c., Cheng, T.-L., Chen, Y.-J., Natsagdorj, S. and Chiang, J. Y. (2015). TCM tongue diagnosis index of early-stage breast cancer. *Complementary Therapies in Medicine*. 23(5), 705–713. ISSN 0965-2299. doi:10.1016/j.ctim.2015.07.001. Retrievable at https://www.sciencedirect.com/science/article/pii/S0965229915001077.

Lo, L.-c., Hou, M., Chen, Y.-l., Y. Chiang, J. and Hsu, C.-c. (2009). Automatic Tongue Diagnosis System. In *Biomedical Engineering and Informatics*. November. 1–5. doi:10.1109/BMEI.2009.5304910.

Mansour, R. F., Althobaiti, M. M. and Ashour, A. A. (2021). Internet of Things and Synergic Deep Learning Based Biomedical Tongue Color Image Analysis for Disease Diagnosis and Classification. *IEEE Access*. 9, 94769–94779. ISSN 2169-3536. doi:10.1109/ACCESS.2021.3094226. Conference Name: IEEE Access.

Moeskops, P. (2022). *Deep learning applications in radiology: image segmentation*. Retrievable at https://www.quantib.com/blog/medical-image-segmentation-in-radiology-using-deep-learning.

Neelam, S. (2021). *YOLO for Object Detection*. Retrievable at https://bit.ly/3XUuoNO.

Pradhan, S., Zainuddin, A. A., Sahak, R. and Yunos, M. F. A. M. (2022). Investigation into Smart Healthcare Monitoring System in an IoT Environment. *Malaysian Journal of Science and Advanced Technology*, 73–77. ISSN 2785-8901. doi:10.56532/mjsat.v2i2.53. Retrievable at https://mjsat.com.my/index.php/mjsat/article/view/53.

Ronneberger, O., Fischer, P. and Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. In Navab, N., Hornegger, J., Wells, W. M. and Frangi, A. F. (Eds.) *Medical Image Computing and Computer-Assisted Intervention MICCAI 2015*. (pp. 234–241). vol. 9351. Cham: Springer International Publishing. ISBN 978-3-319-24573-7 978-3-319-24574-4. doi:10.1007/978-3-319-24574-4_28. Retrievable at http://link.springer.com/

10.1007/978-3-319-24574-4_28, series Title: Lecture Notes in Computer Science.

SINGH, R. (2018). *What is Base64 Encoding and How does it work?* Retrievable at https://www.base64encoder.io/learn/.

Subandi, M. H., Kamarudin, N. D., Yusof, M. A. and Bakar, A. A. (2019). Prototyping Digital Tongue Diagnosis System On Roborealm And Raspberry-PI. *Zulfaqar Journal of Defence Science, Engineering & Technology*. 2(1). ISSN 2773-5281. Retrievable at https://zulfaqarjdset.upnm.edu.my/index.php/zjdset/article/view/14, number: 1.

Synopsys (2023). *What Is Medical Image Segmentation and How Does It Work? | Synopsys*. Retrievable at https://www.synopsys.com/glossary/what-is-medical-image-segmentation.html.

Tania, M. H., Lwin, K. and Hossain, M. A. (2019). Advances in automated tongue diagnosis techniques. *Integrative Medicine Research*. 8(1), 42–56. ISSN 2213-4220. doi:10.1016/j.imr.2018.03.001. Retrievable at https://www.sciencedirect.com/science/article/pii/S2213422017302603.

Tarute, A., Nikou, S. and Gatautis, R. (2017). Mobile application driven consumer engagement. *Telematics and Informatics*. 34(4), 145–156. ISSN 0736-5853. doi:10.1016/j.tele.2017.01.006. Retrievable at https://www.sciencedirect.com/science/article/pii/S0736585316307006.

Techopedia (2017). *What is Bitwise Operator? - Definition from Techopedia*. Retrievable at http://www.techopedia.com/definition/3467/bitwise-operator.

TensorFlow (2022). *Convolutional Neural Network (CNN) | TensorFlow Core*. Retrievable at https://www.tensorflow.org/tutorials/images/cnn.

TensorFlow (2023a). *Save and load models | TensorFlow Core*. Retrievable at https://www.tensorflow.org/tutorials/keras/save_and_load.

TensorFlow (2023b). *TensorFlow Lite | ML for Mobile and Edge Devices*. Retrievable at https://www.tensorflow.org/lite.

Wang, Z., Li, H., Yue, X. and Meng, L. (2022). Briefly Analysis about CNN Accelerator based on FPGA. *Procedia Computer Science*. 202, 277–282. ISSN 1877-0509. doi:

10.1016/j.procs.2022.04.036. Retrievable at https://www.sciencedirect.com/science/article/pii/S1877050922005701.

Wu, K. and Zhang, D. (2015). Robust tongue segmentation by fusing region-based and edge-based approaches. *Expert Systems with Applications*. 42(21), 8027–8038. ISSN 0957-4174. doi:10.1016/j.eswa.2015.06.032. Retrievable at https://www.sciencedirect.com/science/article/pii/S0957417415004340.

Zhou, C., Fan, H. and Li, Z. (2019). Tonguenet Accurate Localization and Segmentation for Tongue Images Using Deep Neural Networks. *IEEE Access*. 7, 148779–148789. ISSN 2169-3536. doi:10.1109/ACCESS.2019.2946681. Conference Name: IEEE Access.

Appendix A Project's Github Repository

The code for this project is hosted on Github, the link for the repository is shown in Table A.1 along with description of the files.

Table A.1 Github repository and file description

File name	Description
(Repository link)	`https://github.com/amdgaafar/tongueDiagnosis-rpi`
main.cpp	C++ programm file that contain the segmentation, coating removal and classification functions
main-rpi4	The executable file, compiled for the Raspberry Pi 4 with ARM architecture 64-bit
api	Directory that contains the Flask server used for the communication with the mobile application to send and receive data
api/ unet_tongue_segmentation.h5	The U-Net model file that contains all the weights which is saved after training and imported during execution
api/app.py	Source code for the Python script that calls the main-rpi4 for the threshold-based segmentation and loads the model for the U-Net based prediction