DOGODO: IOT BASED ENHANCED MOBILE APPLICATION TO PROVIDE ESSENTIAL HEALTH SERVICE TO DOGS

2021- 162

T. S. Chethana Fernando

IT18001730

B.Sc. (Hons) in Information Technology Specializing in Information Technology

Department of Information Technology

Sri Lanka Institute of Information Technology

Sri Lanka

October 2021

DOGODO: IOT BASED ENHANCED MOBILE APPLICATION TO PROVIDE ESSENTIAL HEALTH SERVICE TO DOGS

T. S. Chethana Fernando

IT18001730

Dissertation submitted in partial fulfilment of the requirement for the Degree of BSc (Hons) in Information Technology Specializing in Information Technology

Department of Information Technology

Sri Lanka Institute of Information Technology

Sri Lanka

October 2021

DECLARATION

We declare that this is our own work, and this dissertation does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any other university or Institute of higher learning and to the best of our knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Name	Student ID	Signature
T.S. Chethana Fernando	IT18001730	

The above candidate has carried out research for the bachelor's degree Dissertation under my supervision.

…… …….

Signature of the supervisor Date

(Ms. Disni Charuka Sriyarathna)

…… ……..

Signature of the co-supervisor Date

(Ms. Shalini Rupasinghe)

ABSTRACT

Skin is the largest and most important organ in a dog's body. It acts as a barrier to protect the dog's body from infections, parasites, and elements. It maintains the body's internal environment and prevents moisture loss, and the skin can be 12% to 24% of body weight, depending on the species and age. The skin is located on the outside of the dog's body. It is easily exposed to the environment and is susceptible to injuries and diseases. Dogs can be infected with different skin diseases. Some guardians will try to treat their infected dog on their own if the dog has been infected with certain diseases. This happens cause dog owners do not have time to take their dogs to a veterinarian when they are unwell. However, sometimes this treatment was not appropriate for that particular problem and can cause it to become worse. Everyone tries to accomplish things more efficiently in today's hectic environment. With the use of information technology, everything provided to people are in an electronic way. While most dogs' skin problems are not emergencies, it is vital to analyze how this condition can be treated accurately. Computer vision applications have been developed to identify images patterns rapidly. In the past years, efficient methods developed with machine learning and deep learning techniques have provided impartiality and high accuracy in detecting skin diseases. One of the main goals of our study is to develop a concept that focuses on diagnosing six common dog diseases using machine learning and deep learning. Here, we considered the skin diseases of Hair loss, Hot spots, Lumps, Rashes, Swelling Ticks, and wounds, as well as healthy dogs. This research proposes a feasible solution for dog owners. Thus, developed an online dogs' diseases detection system that used the concept of the expert system to users in detecting diseases and provided popular markets treatments with useful suggestions via a chatbot with a mobile-first approach-based web application.

Keywords – Skin Diseases, Computer Vision, Machine Learning, Deep Learning, Chatbot, Mobile-First Approach

ACKNOWLEDGEMENT

First of all, I would like to give my heartfelt gratitude to our supervisor Mrs. Disni Charuka Sriyarathna for all guidance, support, and motivation. She was very helpful and generous with her valuable time.

Dr. Janaka, the lecturer in charge of the research project module, deserves special thanks for his direction and continuous monitoring, as well as for supplying essential project information and for their help in finishing the project.

We would also like to thank the personnel and professors at the Sri Lankan Institute of Information Technology, who have been invaluable guidance and support over the four years. Finally, to encourage and strengthen our family and friends, who have always been cornerstones of our achievement.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviation	Description
ANN	Artificial Neural Network
CNN	Convolutional Neural Network
CSC	Color Space Conversions
DL	Deep Learning
HSL	Hue Saturation Lightness
HSV	Hue Saturation Value
ML	Machine Learning
NPL	Natural Language Processing
NUL	Natural Language Understanding
OpenCV	Open-Source Computer Vision
POS	Parts-of-Speech
RGB	Red Green Blue
SVM	Support Vector Machine
JSX	Java Script XML
XML	Extensible Markup Language

1 INTRODUCTION

1.1. Introduction & Background Literature

The Dogs are the best friend of men anywhere in the world, particularly to real pet enthusiasts. Dogs understand their owners' feelings and are open to their owner's emotions. There are only a couple of the explanations that most people choose to have a dog as a friend than some other pet. Other aspects of general care and feeding and exercising the dogs are needed to maintain the pet dogs' healthy throughout its life. Routine veterinary care, including vaccinations, skin disease control, and dental care, dogs grooming, and protection from household threats, are among them. A total of four major capabilities have been integrated into the suggested solution of mobile-first approach with a web-based application. The system mainly focuses on internal health, voice recognition, emotional transformation, skin disease, breeding patterns, and pet breeding outcomes. Figure 1.1 shows As per the above high-level architecture diagram, the system will contain four main functions of the DOGODO, the mobile-first approach with a web-based application.

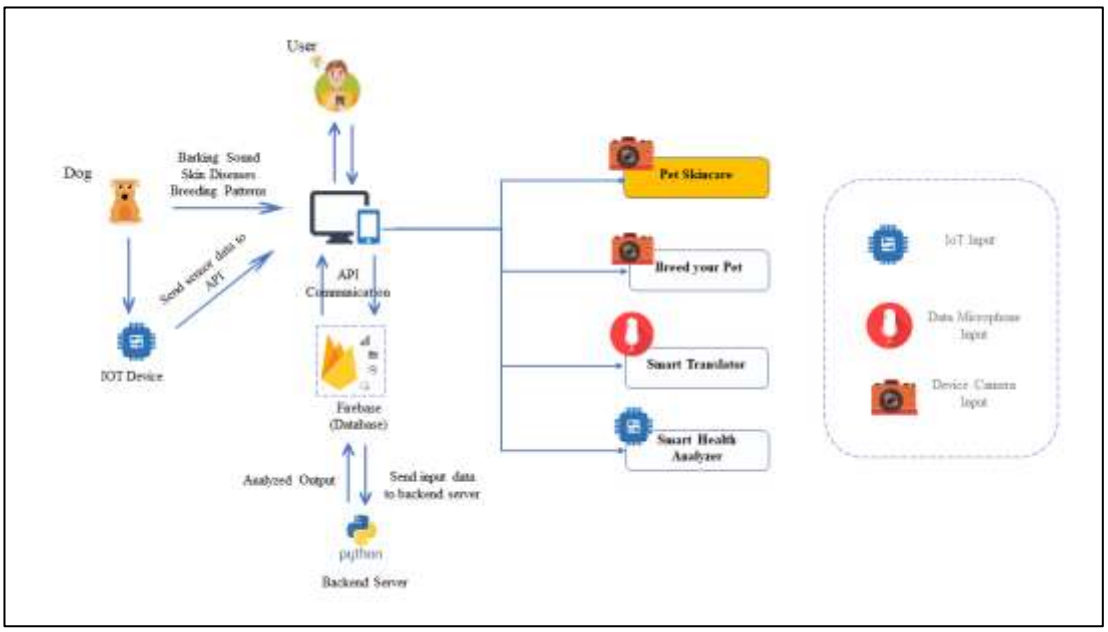


Figure 1.1- High-Level Architecture Diagram

Here, we mainly consider highlighted component in high level architecture diagram. Most pet owners probably overlook the importance of a pet dog's skin coat. The kindest and most responsible thing pet owners can do for him is to provide appropriate health care. Knowing about common skin dog diseases and how to avoid and treat diseases can make it much easier for pet dog owners to provide that care. The skin is a complex structure that provides an essential function in a dog's overall health. However, it is undeniable that these lovable animals are to various skin diseases, especially their skin, as it is the most extensive and most accessible organ of skin in their body. Comparable to humans, unfortunately, skin diseases are prevalent in dogs [1].

According to a local veterinarian, dogs have two main types of skin diseases: minor and severe skin diseases. The severe ones range from acute to self-limiting issues, while the incurable one is known as a persistent condition needing lifelong care. Many minor skin diseases originate in dogs, the most common inherited skin disease [2]. For example, contagious skin diseases include parasitic, bacterial, fungal, and viral skin diseases. The non-contagious skin diseases include mange, mites, and lice, all fall within this category, along with flea and tick infestations [1], [3]. Despite technologies, there will still be limited use of the proposed system to identify skin diseases in dogs, which are not properly localized. The current proposed system for diagnosing skin diseases in dogs can only identify two types of skin diseases.

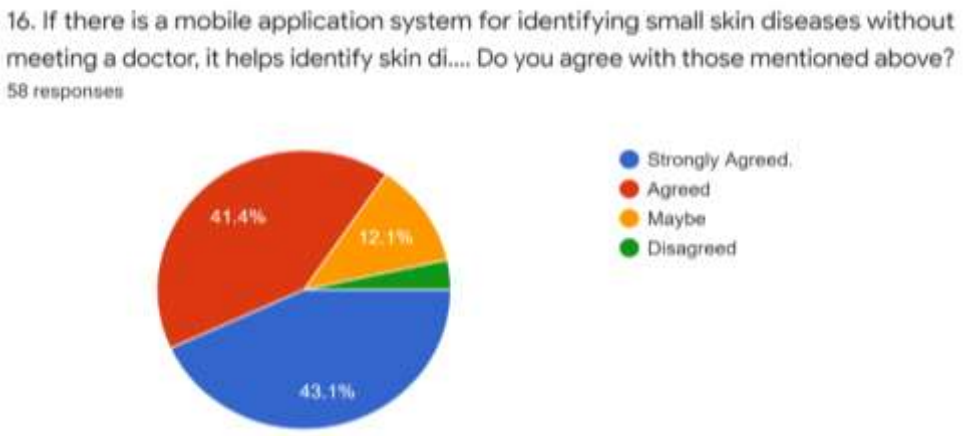


Figure 1.2- Summary of responses for mobile-first application

According to local dog owners who took part in the survey, dogs with skin diseases are underestimated, and their skin diseases are ignored due to a lack of time to see a veterinarian. Further, many dog owners have limited knowledge of dog skin diseases. Therefore, most dogs' owners agree with having a mobile-first approach with a web-based solutionTherefore, this study proposes developing a web-based application for dog owners and introducing a chatbot to assist dog owners in finding veterinarians about dog skin problems and finding popular market products for skin diseases. With this, dog owners can take an image of their dog's skin disease using a camera on the back of an Android-powered phone and then use the image gallery to upload it to the web-based application to detect pet dog skin diseases. If it is a minor issue that popular products can heal in the market, it will pop up via the chatbot in the web application. (for example, powders for swollen skin resulted from tick bites, dog shampoos for dry skin, rash creams, Etc.). The user is informed through the chatbot in the web application that their pet dog needs medical treatment for a severe problem. Features of the web application allow the user to find veterinarians and veterinary centers in the nearest city. Moreover, the chatbot provides further operations information on dog skin diseases and a conversational approach to accurately diagnosing pet dog skin diseases. The research study component proposes to offer a feasible solution for dog owners.

Many researchers have suggested image-based techniques to classify pet dogs skin diseases. Hereabouts, we quickly study some of the technologies and techniques mentioned in the literature review. In this research, one of the sub-components of pet skincare focuses on identifying skin diseases in dogs, and the study examined current solutions using mobile-first approaches web application platforms and mobile application platforms. The official websites of current solutions and published research papers from existing studies were reviewed throughout this search, and selected applications were reviewed to determine the features of the solutions that were provided. A literature survey was conducted before the system's design phase to identify and examine existing systems. From hosted apps and the Google Play Store, the pet skincare component sorted out the required apps. Existing commercial

applications and studies were taken into considerations when evaluating the applications.

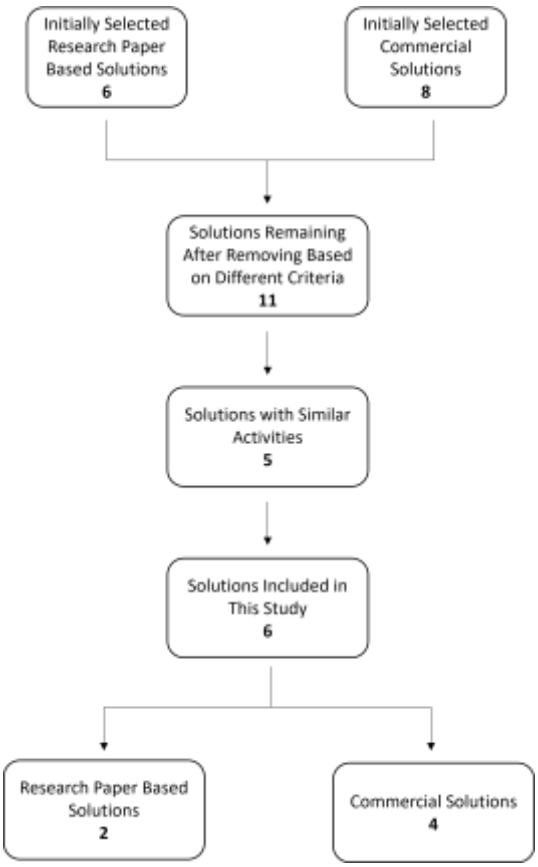


Figure 1.3 - Literature review Screening Process for proposed system

In this proposed study mainly consider about,

✓ **What** are the image processing algorithm and the taks of image processing **?**

When an algorithm analyzes images to identify data insights or support automated tasks in computer vision use cases, this is referred to as image processing. The proposed method utilizes the four tasks illustrated below diagram to identify skin diseases.

- Image pre-processing
- Feature extraction
  - ❖ Color Image Processing using RGB, HSV, HSI, HSL

- ❖ Morphological Processing using canny edge with threshold g-f
- Segmentation
- Image classification and identify the image.

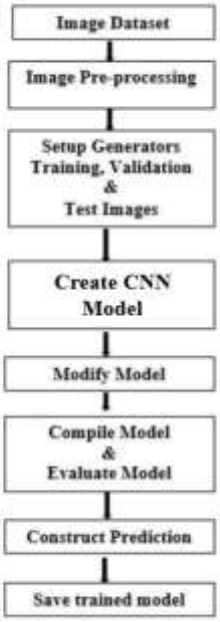


Figure 1.4 - Image Processing
Systamatic flow chart

This approach is suggested to identify skin diseases using color representations without veterinarians' involvement. After gathering the data need to be pre-processed. Image pre-processing refers to operations on images at the most basic level. Outlier detection, missing value treatments and removing unwanted or noisy data are all part of data pre-processing. Pre-processing focused on enhancing image data by reducing unwanted distortions or improving certain important image features for future processing and analysis tasks. After that, we need to decide what algorithm we can use to identify the pet dog skin diseases. In machine learning and deep learning, there are many algorithms for image processing. Here, mainly using CNN algorithm for image classification, K-means algorithm for image segmentation, and colour gradients techniques such as RGB, HSV, HIS, HSL, Morphological edge threshold for feature extraction to recognize diseased skin in dogs, and the second implements [4], [5].

The following image processing step is feature extraction. Feature extraction is a kind of dimensionality reduction used in image processing. Many variables are a feature of these large data sets, which require a huge amount of computer resources to analyze. The main objective of feature extraction is to extract the most important data from the original data and represent it in a lower-dimensional space. The feature extraction method helps minimize the number of processing resources required without sacrificing essential or relevant data. For the feature extraction, there will be used two techniques.

Almost every computer system, TV, and film use RGB because it is very simple to apply. RGB is not a suitable constructive option for color image processing because of important perceptual nonlinearity via visual observation, Machine dependence, and chrominance and luminance data integration. As a result, it requires a variety of color spaces that are well-suited to processing. A conical HSV color space is used to convert RGB to HSV. The RGB to HSV conversion algorithm is described in the literature [5]. R, G, and B, which were inputted, chose the maximum and minimum. Using the saturation and value settings, it can choose and modify the desired hue. The HSL and HSV color spaces are virtually identical, except that HSL gives high color values to colors that are close to white and have a limited saturation. This feature would raise the model's complexity level. The Hue Saturation Intensity (HSI) color model is similar to human vision's color detecting characteristics. Converting from RGB to HSI or back is more complicated than with other color models [5].

Morphological features are extracted from segmented image forms using both boundary and area-based techniques. Image-boundary data is used in both kinds of techniques, while area-based methods additionally utilize image-interior points. Area-based techniques are more resistant to minor changes in image shape and are simple to implement. Grayscale images with pixel values ranging from 0 to 255 are used for thresholding. When classifying pixels in an image, establish an upper and lower limit for each group [6].

The unsupervised K-Means clustering method is used to identify the interest region from the background. Based on the K-centroids, it clusters or divides the supplied data

into K clusters or segments. Converting an image into a collection of pixel areas represented by a mask or a labeled image is the process of image segmentation. It may process just the relevant parts instead of the entire image by splitting it into segments. The K-means algorithm attempts to reduce the square error in the following object function. The goals' of the K-means function is as follows:

$$J = \sum_{j=1}^{c} j \sum_{i=0}^{n} i \left\| z_i^{(j)} - v_j \right\|^2$$

Figure 1.5 – K-means algorithm [4]

where $\left\| z_i^{(j)} - v_j \right\|^2$ is the chosen distance measure between every point, $z_i^{(j)}$, and the cluster, v_j [4]. The value of this function represents how close the data clusters n is to their cluster models. The steps of the algorithm are as follows:

I. For the space comprising the grouped items, choose the k mark. These are the designs for the first group.
II. Assign each object to the team that has the most similar prototype.
III. Recalculate the positions of the k prototypes once all objects have been allocated.
IV. Repeat steps II and III until the prototype values do not alter any more. As a result, items are divided into categories, and the metric to be reduced will be computed.

In this system, the higher the number of characteristics from an image, the higher the system's accuracy [7]. The focus here is on analyzing the different segmentation methods that can be used to classify different skin diseases using image processing. A mechanism of disintegration is defined, and it falls on the boundaries of the infected site to add more features [7]. This process focuses on sophisticated algorithmic databases and images from various pet dog skin tools and suggests developing a

technique for dark skin diseases. In this case, the numerous pet dog skin diseases listed above may lead to severe issues by spreading to different regions. Therefore, the proposed system automatically determines the severity of skin diseases.

The system uses Computer Vision to identify and interpret images of captured skin diseases and the trained CNN model to determine the skin disease and its classification [8]. To improve object detection, both TensorFlow and Flask will be used together. The object defined with OpenCV is not optimal and utilizing TensorFlow extends the reach of network and algorithm research. TensorFlow is designed for data access and handling, while TensorFlow is optimized for data access and handling. As a result, these are used to detect objects together [8], [9]. The new technique is proposed to identify pet dog skin diseases using a proposed system combined with machine learning and deep learning techniques. The suggested CNN algorithm would be included in the current study [8]. CNN's procedural phases are represented in **Figure** 1.4. The CNN reduces the scale of the input images to make them easy to process and identify the skin diseases image.

- ✓ **What** are the techniques used for the chatbot and the task of the chatbot.

Chatbots are a computer-aided program that mimics user activity on one side of a chat conversation. They are simulation systems that pretend that the two are engaged in a conversation. They serve as a model for efficient and intelligent relationships with the user on the other end. This functionality is particularly useful for chatbots that answer a large number of queries throughout the day. This is an excellent technique to use if your answer rate to these questions seems to be low and use an innovative spin. The conversational capabilities of a chatbot that operates under a set of rules are limited. It can only respond to a limited number of requests and has a limited vocabulary, and its knowledge is limited by its programming code. An automated banking bot, for example, is a limited bot that asks the caller a set of questions to identify what the caller wants to be done.

In this proposed system to create chtbot using the NLP.js with React freamework. Natural Language Processing (NLP) is a discipline that combines linguistics,

computers, and artificial intelligence [10], [11]. A Natural Language Processor is usually included in these applications. Its purpose is to extract the interactions and intentions and related information and metadata from a piece of plain natural language and convert them into something that a computer can understand. Stemmers are used in NLP.js to increase accuracy and reduce the number of training utterances required to get the same result. The accessibility of NLP as an open-source library increases exposure and comprehension of low-level natural language processing. It would allow technical people to understand better how the conversation is processed to manage language-specific techniques and reach the desired level of accuracy [12].

In order for the system to comprehend the expression, it must first comprehend the syntax of the phrase. Knowing the elements of each word spoken in the statement is one way to do this. Each word's grammatical weight is decided, and then they are all evaluated to see how dependent they are on one another. This is the most crucial stage, as it is at this point, the most reliant term is extracted, and the system's purpose is revealed. It is possible that the knowledge-based user did not send the exact wording. It may mean the same thing as the preceding statement, so it is written differently. Contemporaries and matching sentences may also be determined when balancing this type of synonymous phrase [12].

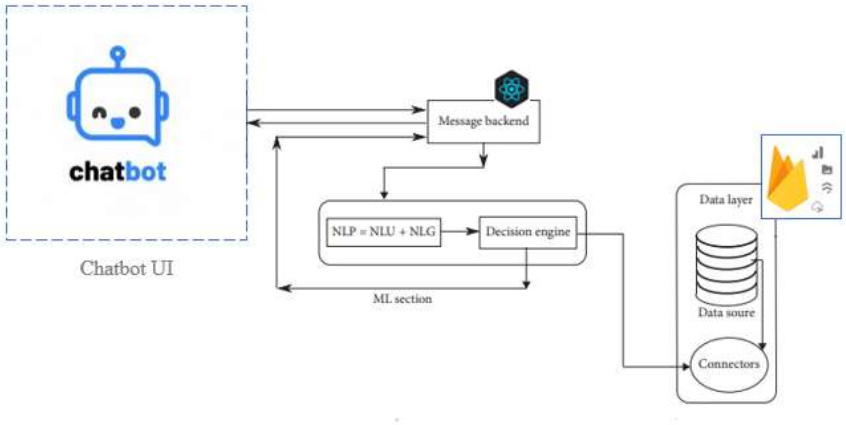


Figure 1.6 - Chatbot System Architecture

The study proposed after identifying the skin disease, the user can start a conversation with the chatbot. The following are some of the features of React-based chatbot:

- Create a basic conversational chatbot by creating a process. A JSON object may describe all of the stages.
- Tokenization, and other basic NLP tasks.
- The chatbot can change the chat's appearance.
- The options phase provides several options for the user to pick from, and the bot will go to the next step depending on the user's selection.
- The custom phase is where we render our own React.
- Update phase where specify which step should be changed after the user completes the current step.

Render a scrollable list depending on the data received in the API response in response to the user's input on a particular stage. The proposed systems' chatbot depends on identified the skin disease the application, getting that skin diseases name into the chatbot, and it allows the end-user to find skin diseases treatments and search for the veterinary-related areas using Google maps API.

1.2 Research Gap & Existing Solutions

Within the proposed study, a research gap revealed the concerns that any existing studies have not addressed. During the literature review, I have found several ways to detect dogs' skin diseases and analyze systems. Hereabouts, the relevant research on existing skin disease systems is considered and compared.

- **Skin Infection Detection in Android using Image Processing**

The software helps people to be more knowledgeable of their skin health status to recognize skin diseases. Infected skin must be collected to recognize and scan the captured image utilizing the OpenCV image processing feature. The system then tracks the signs and effects, evaluates and care for the skin's contaminated region. Of these, utilizing that, only two forms of skin diseases can be described [1].

- **Segmentation and Classification of Skin Lesions for Disease Diagnosis**

This study included five separate skin infections: seborrheic keratosis, melanoma, bullae, squamous cell carcinoma, and shingles. The fragmentation procedure is done by filtering the skin image and absorbing the injured regions to eliminate external noise and hair. The application of image fragmentation was also used in the analysis to classify the injured areas with the derived color and texture [7].

- **Design and Development of Online Dog Diseases Diagnosing System**

This study focuses on common dog diseases and aims to implement a platform in the future that will detect dog diseases such as parvo, distemper, glaucoma, jaundice, colitis, whooping cough, virus, and leptospirosis. The research demonstrates the importance of early detection of the above diseases. Furthermore, the device assists dog owners in identifying the condition and including clinical instructions regarding outcomes and advice [3].

- **Skin Cancer Detection Using Artificial Neural Networking**

The study aimed to create an automatic application that uses images of skin lesions taken with a digital camera to predict a patient's likelihood of developing melanoma. The study concentrated on the benefits of neural networks in the analysis of skin cancer pictures. ANN has been used in cancer diagnosis to overcome issues that cannot be solved by other recognition methods or conventional image recognition [14].

- **M-Health Skin: Disease Analysis System Using Smartphone's Camera**

The analysis showed a smartphone-based technology that helps users to identify the skin for skin diseases. M-Health solution is a smartphone-based skin health analysis system requiring mobile neural networks to identify regular and irregular skin images. An intelligent learning algorithm and a mobile phone's camera are used [15].

- **Automating skin disease diagnosis using image classification**

Skin cancer incidence is progressively rising, especially in Caucasian population countries that inspire this research. The study aims to minimize the dependence on doctors' opinions by making usage of a feature based on texture analysis and diagnosing the lesion by using the artificial neural network as a classification technique. The segmentation and classification method, automatic skin cancer detection is possible and attainable. While there have been numerous successes in recent advancements in medical diagnostic automation, this research focuses on the wide availability of devices [13].

The proposed system displays the detected skin diseases and analyses the severe and minor skin diseases of the common dogs' skin diseases via a mobile-first approach with a web-based application. Moreover, the end-users can view the probability of the skin diseases accuracy with other skin diseases. Furthermore, provide a conversation with a chatbot without paying any cost. Therefore the chatbot is cost-effective. Predicting the popular medicine products in the market (for example, powders for swollen skin resulted from tick bites, dog shampoos for dry skin, rash creams, Etc.) for the minor diseases and the owners have the option to download treatments images and view the instruction of the treatments via chatbot. Moreover, the owners were notified that their pet dog needs veterinary attention and can find the veterinary hospital/centers through the chatbot system.

Table 2.1 - Comparison of Existing Solutions

	Skin Infection [1]	***Skin Lesions [7]***	***Dog Diseases [3]***	***M-Health Skin [14]***	***Proposed System***
Identifying different comman eight dog's skin diseases.					
Identiying the non-infected dogs.					
Provides a conversation with the chatbot without paying any cost.					
Show the proberlity of the skin diseases accuracy.					
Prediction of the treatment for skin diseases.					
Visible to download treatments image throuth the chatbot					
Find veterinary related areas.					

2 RESEARCH PROBLEM

The development of health care has grown reliant on information technology and computer science. People are attempting to examine deeper into the many sectors of the health domain using computers and information technology to discover answers. On average, we are living in a golden age of medical research. People are always eager to get involved in this area, and they come up with new ideas to innovate human life. A thorough history, physical examination, and appropriate diagnostic tests are required to diagnose the causes of a pet dogs' skin condition accurately [1], [15].

Many skin diseases have identical symptoms, making an immediate diagnosis difficult. A veterinarian may prescribe various laboratory tests based on the pet dog's history and physical examination. Diagnostic tests include skin scrapings and hair microscopic analysis, hair or skin swab cultures, specialized skin testing, blood, and urine tests, and even biopsies [3]. The results of the laboratory tests may take several days to arrive. A proper diagnosis typically requires more than one visit. One of the research study's main problems is targeting to identifying common skin diseases in canines. Usually, the dog's skin diseases can be caused by environmental factors, food, insect bites, or various dog activities. Most people lack the proper knowledge about the skin diseases of dogs. Most of the time, these skin diseases can be easily treated with over-the-counter medication [1], [3]. Often, canines develop skin diseases that need immediate relief.

The data received from the first survey is presented in the following figure 2.1. If you look closely, you will notice that most pet dog owners cannot understand if their pet dog has an illness. When we get it as a percentage, it was 84% of survey respondents. Figure 2.2 shows the extent to which a pet owner can understand a dog's illness. Most of the respondents cannot properly understand the dogs' illness. The main reason for those two problems is that they do not have proper knowledge about their pet dogs or pay attention properly to their pet dogs.

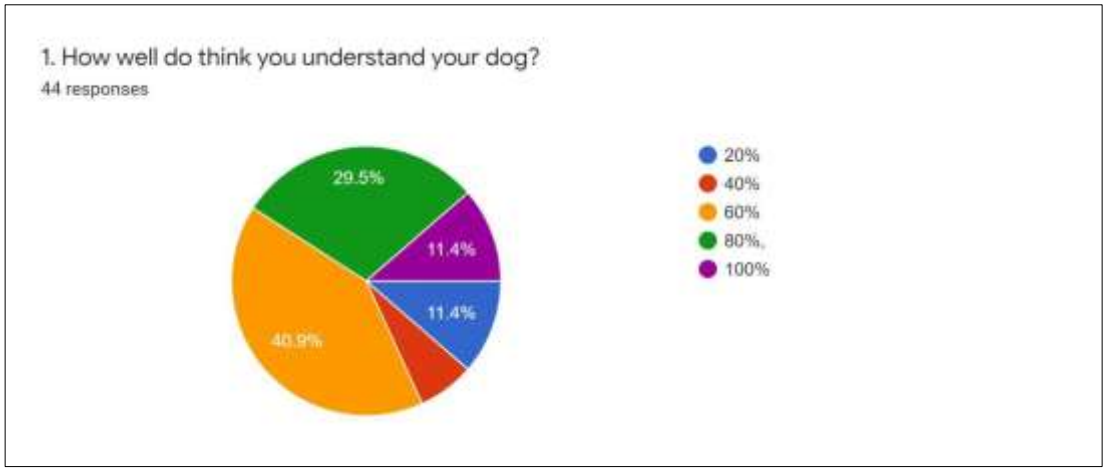


Figure 2.1 - Summary of responses regarding understanding the dog behaviors

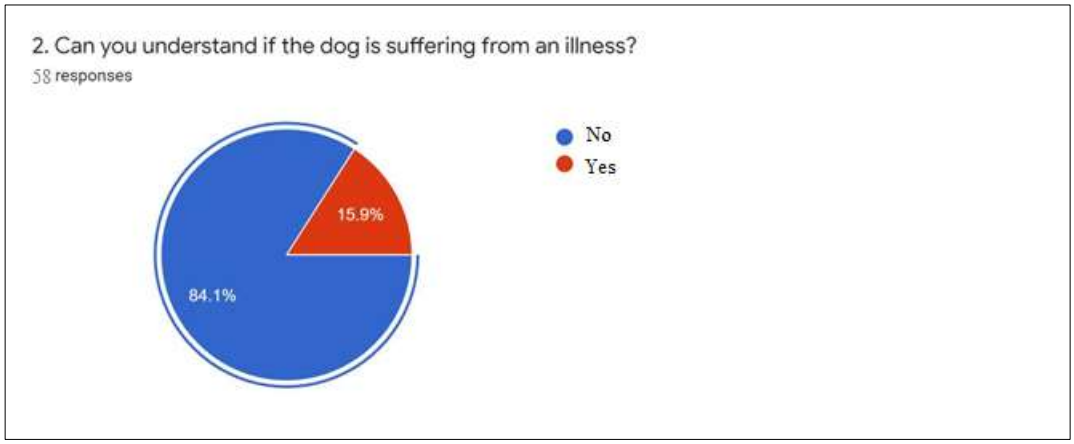


Figure 2.2 - Summary of responses regarding understating the dog illness

However, veterinarians' availability and the costs associated with the hospital visit delay the diagnosis and treatment. This is especially an issue if the dog is abandoned or stray [15]. The most common solution to skin problems is for owners to take their pet dog to the pet. However, in some areas, veterinary services are not available 24 hours a day. Such that, the Lack of alternative treatments directly affects the dog owner and the dog.

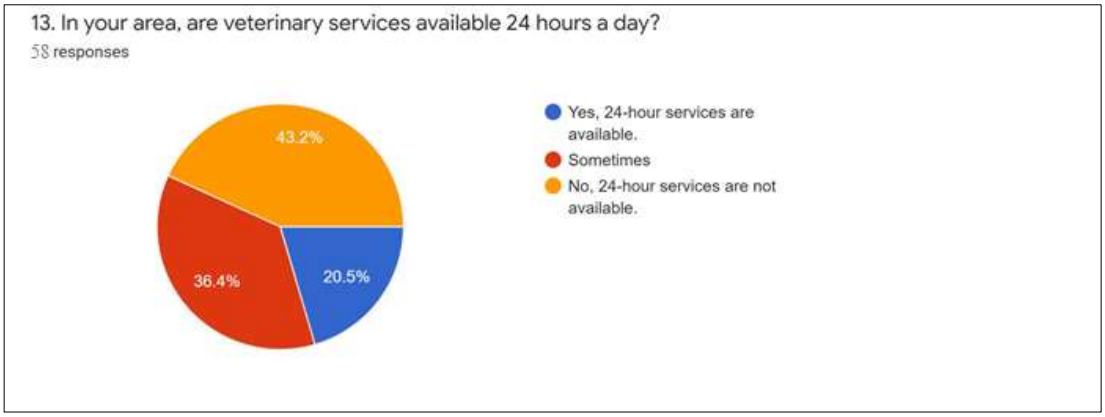


Figure 2.3 - Summary of responses regarding 24-hour veterinary service

In some cases, dog owners abandon the dog based on their health and behavioral issues without knowing how to react to their disease-related issues. The number of stray dogs has gradually increased due to the repeated action of people abandoning dogs. We can see a lot of such problems in recent times. The conclusion from these facts is that the existing ways to identify skin diseases for dogs are inefficient and not systematic. If users are busy, they may not have enough time to concentrate on the dog 24 hours a day. As shown in Figure 2.4, more than 51% of survey respondents say they do not have enough time to focus on the dog 24 hours a day.

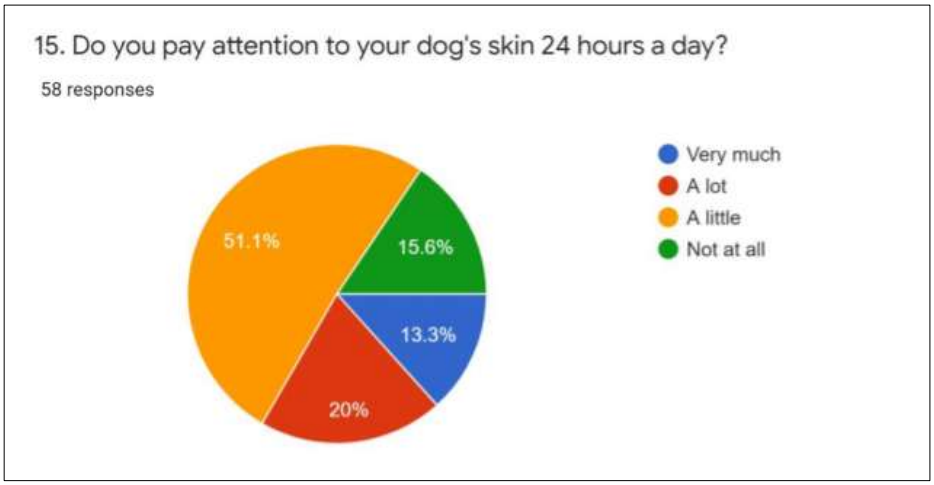


Figure 2.4 - Summary of responses regarding pet owner's attention

Therefore, users are looking for the easiest way to take care of their pet dogs. Because, for absolutely anything today, there is an app. As shown in Figure 1.8, more than 42% of survey respondents strongly agree with a localized mobile app chatbot for identifying skin diseases.

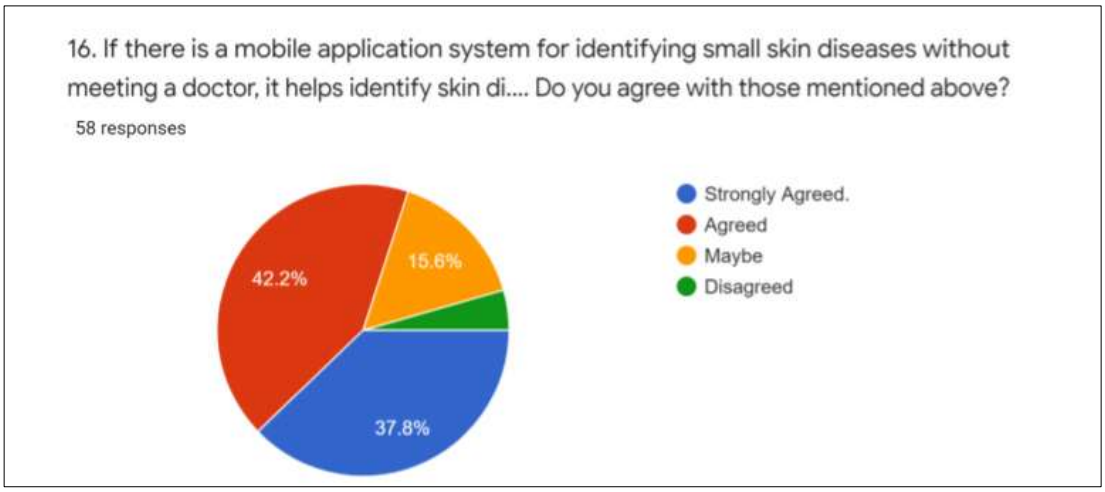


Figure 2.5 - Summary of responses regarding the mobile-first application

The data received from the first survey is presented in the following figure 2.5. It showed that the majority of pet owners are not used mobile applications. With this COVID pandemic situation, many people are looking for solutions to make day-to-day life more manageable with mobile phones/tablets/laptops/computers. In such a situation, the proposed system to identify dog skin diseases without the help of a doctor is a perfect solution.

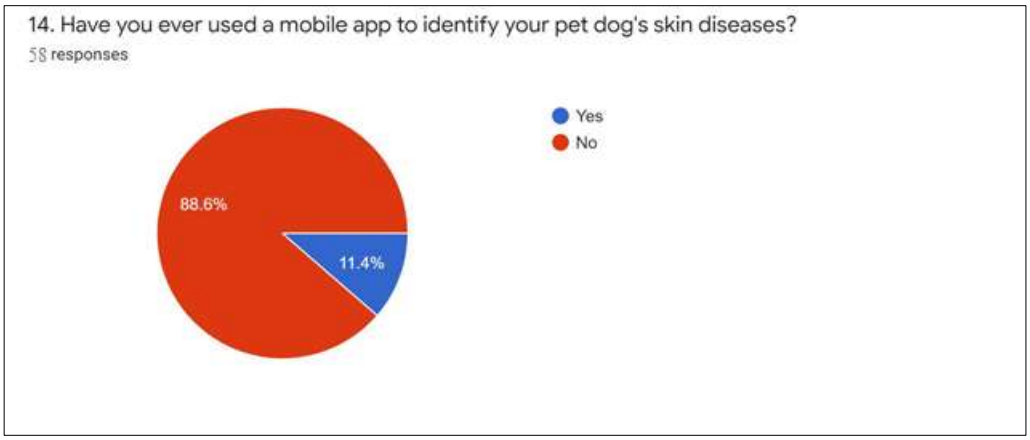


Figure 2.6 - Summary of responses regarding used of mobile application

3 Research Objectives

3.1 Main Objective

The component's main purpose is to introduce a technology-based mobile-first approach to dog owners to identify different skin diseases in pet dogs. This component will use preprocess images from reliable resources to compare device upload images to predict skin-related disease accurately. Therefore, only the mobile camera should be used to detect skin issues, and this component is useful for users. Based on image observations, the app will determine whether skin-related diseases are minor or severe. Therefore, dog owners can get to the best conclusion about their dog's skin diseases.

3.2 Specific Objectives

The following specific aims can be obtained in the research study to gain a high precision dog skin detector and the chatbot.

- **Data preprocessing**

Once the dataset is collected, we have to pre-process this data. When looking at a computer, most real-world data is not formatted correctly, and there are many flaws. In machine learning and deep learning, data preparation is a separate segment. Missing values and unsuitable formats exist in real-world data. However, before we can do anything with the data, such as create a machine learning model, we must first clean and train them. Data preparation entails evaluating the suitability of these processes. Both the test and the training datasets must be pre-processed independently in order to train both models.

- **Train the skin diseases image dataset**

When training the networks, the collected data set must be split into appropriate training and test set. There are many hyperparameters to evaluate. To build a proper neural network, appropriate values must be given to them, increasing the number of training epochs and finding the best optimizer to maximize accuracy.

- **Evaluate the accuracy and the performances**

After the neural networks have been trained, they must be evaluated using new data. The neural network is evaluated to ensure that it is ready to provide the expected outcomes. By identifying the dogs' skin diseases, the neural network must provide the optimum solution.

- **Optimize the accuracy and the performances**

If the networks are not performing well after evaluation, then parameters should be modified and changed. It can be improved both accuracy and efficiency.

- **Identifying the skin diseases**

After the evaluation and the optimization, using the saved model predict the skin diseases uploading image via mobile-first approach with a web application and display the pattern of the uploaded skin disease and the display the name of the skin disease with how much the diseases match with the train dataset. Here, we consider the pet dogs' skin diseases of Hair loss, Ticks, Lumps, Rashes, Hot spots, Swellings, Dry skin conditions and wounds. If the dog is healthy, the proposed system will be displayed it as a healthy dog.

- **Implementation of chatbot**

This study attempts to overcome this ongoing challenge by developing a chatbot to predict skin diseases treatments and the related veterinary centers. The chatbot's purpose is to have a smart, accurate, and real-time conversation with the pet dog owners. The chatbot will be accessible through portable mobile devices or computers, enabling pet owners to access it from anywhere and anytime, providing 24-hour online service.

- **Prediction of Popular products for minor issues**

If it is a minor issue that popular products can heal in the market, train a model for the product suggestion. It will pop up through the chatbot. (for example, powders for swollen skin resulted from tick bites, dog shampoos for dry skin, rash creams, Etc.).

- **Prediction of veterinarians related to the area**

The user is informed that their pet dog needs veterinary attention for the severe issue through the app chatbot. There will be a training model with details about the veterinary surgeons and veterinary centers for this suggestion via google map. Features of the application allow the user to find veterinarians and veterinary centers in the nearest city.

- **System Integration & Deployment**

After building both models, we must create separate REST endpoints that accept diseases as input and provide the detection pattern. We also receive the treatment images as output and may view the treatment instructions. At the end of the above, we must deploy the models using Python, including implementing and training the models. The API can use these models to make predictions. To deploy the API, we can use open-source Firebase services.

4 METHODOLOGY

The proposed approach is a useful tool for determining pet dog's skin diseases and predicting skin disease. A hybrid architecture comprising image processing, machine learning, and deep learning techniques are utilized in this proposed system to quickly identify disease types with potential accuracy. Preprocessing, segmentation, and feature extraction are all stages in the image processing phase. Processing, training, and detection are the three stages in the machine learning phase. Color, texture, and shape characteristics are retrieved from the dog's skin disease input images using the proposed system's HSL/HSV conversion techniques, Morphological image processing, and Thresholding technique Canny edge detection algorithm for feature extraction. The correlation values of the input skin disease image are compared.

An image data collection is segmented into distinct groups or clusters using cluster analysis. The colour space conversion algorithm may be used to complete the conversion process. The suggested approach uses a convolutional neural network for categorization (CNN). Hair loss, Ticks, Lumps, Rashes, Hot spots, Swellings, Dry skin conditions and wounds are among the frequent skin diseases and healthy dogs detected by the classifier model. The use of a neural network for integration produces accurate results. This proposed system analyzes many types of dog skin diseases, saving the user time and cost. For the experimental results of this proposed system, deploy a PyCharm-based Python script. As well as the simple chatbot system, implement with a method in React is implemented in React-based chatbot components that accept input data and return what to display. The JSX syntax is similar to XML, and it is used here. The method may access input data provided into the component.

4.1 System Overview

- OpenCV (Open-Source Computer Vision Library) – Image recognition and optimizing
- TensorFlow – Images preprocessing and classification
- Google Collab – to gain computation power to train the image recognition model.
- Firebase – Database server
- Python – Backend development language
- Flask – Form validation, or any other components where pre-existing third-party libraries provide common functions.
- Keras – Keras acts as an interface for the TensorFlow library
- NLP (Natural Language processing) – For chatbot questionnaire
- React– Development language of the mobile application.

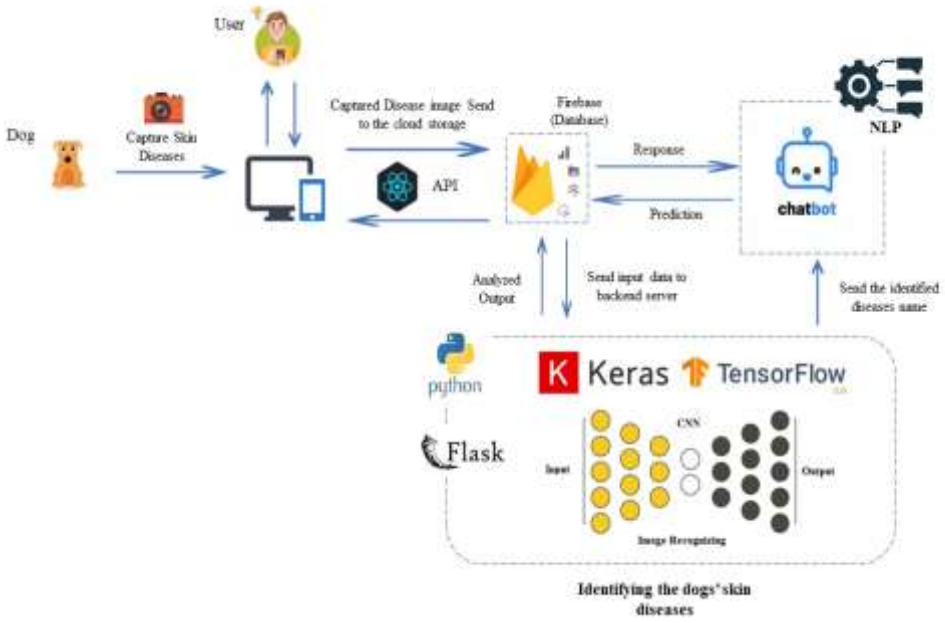


Figure 4.1 - System Overview of the Pet Skincare Component

4.1.1. Data Gathering and Data Pre-Processing

The most important factor in adequately implementing a deep neural network is data collection. Because a stronger data collection is required to train a neural network with the highest accuracy, the network will not perform as anticipated if the data set chosen is inadequate. Therefore, the dataset is divided into train and test datasets. For analyzing the dog skin's disease images, necessary images will be collected with the assistance of two veterinarians from the Animal Hospital, Negombo, and the Animal Clinic, Nittambuwa.

Each category has a varied number of skin diseases images. This study utilized a total of 2 639 images. The number of images in each category is split into training and testing sets. There are 1 924 images for training and 715 images for test dog skin diseases. Apply data augmentation to the dog's skin diseases dataset for image classification by randomly shifting or horizontally flipping images. As a result, we will provide a quick rundown of the methods below. The images are pre-processed and scaled to 225x225 in the training phase.

4.1.2. Feature Extraction

Diseases image enhancement involves intensity and contrast manipulation reduction, background removal, edge sharpening, filtering, and more in image processing. Mathematical Morphology techniques, such as the Top-Hat morphological transform, were used to accomplish the purpose. A non-linear conversion of the RGB colour space is used in this colour space model. The HSV and HSL models have been widely applied in computer vision and diseases image processing for element identification or image segmentation [5]. A multi-stage Canny edge detector is a Canny filter [14]. As well as edge detection has a low error rate, implying that the detection should capture as many of the image's edges as feasible. Image processing, using morphological transformations, is information removal based on size and shape [7].

```
    def process_image(self, image, input_shape) -
> np.ndarray:
        """
        Given a PIL Image, center square crop and resize to
 fit the expected model input, and convert from [0,255] to
[0,1] values.
        """
        width, height = image.size
        # ensure image type is compatible with model and co
nvert if not
        if image.mode != "RGB":
            image = image.convert("RGB")
```

Figure 4.2 - RGB colour model implemetation

4.1.3. Splitting the dataset into features and label

We need to divide the dataset into features and labels after it has been imported. The attributes are referred to as features. It was called independent variables. When predicting the labels, the set of features matrix is used [4]. The dependent variables are labels. Diseases are the attributes of independent variables for the first neural network in our study. We have declared two variables for features and labels in order to train the networks. It is known as X and y. The independent variables are identified by character X, whereas the character y identifies the dependent variables.

4.1.4. Splitting the dataset into training and test dataset

We need to divide our dataset into two parts before training our models. The first part is for training, while the second part is for testing. To begin with, as a training set, it will help in the training of the neural network. The accuracy level of the neural network may be analyzed. Based on the use case, this partitioning may be varied. Normally, the training set contains 80% of the data set's proportion.

4.1.5. Image Segmentation

Image segmentation is an essential task in image processing and computer vision, and it entails recognizing objects or areas in an image with similar features. Image analysis includes a wide range of tasks, including segmentation, classification, and interpretation. Image categorization gives labels to individual pixels based on prior knowledge about the issue under consideration [9]. For each object in an image, image segmentation generates a pixel-wise mask. This method allows us to comprehend features in the image at a much more comprehensive level. Using the pixel values of different objects may be a simple way to split them. If there is a sharp contrast between the objects and the image's background, the pixel values and the image's background will be different. Threshold Segmentation is the term for this method. We may set a threshold value in this case. Pixel values that fall below or above that threshold are classed accordingly.

After that, we can use discontinuity to identify edges and therefore determine an object's boundary. It helps us in recognizing the shapes of multiple objects in a single image. The weight matrix's values determine the convolution's output. It facilitates the extraction of features from the input. According to the findings of the proposed study, selecting particular values for these weight matrices helps identify horizontal and vertical edges. The image has been converted to a 2-dimensional array, as can be seen. Then, on this altered array, fit the k-means algorithm to get the clusters.

4.1.6. K-means Algorithm

One of the most important unsupervised learning algorithms is k-Means [7]. The technique uses a basic and straightforward method to categorize the dog’s skin diseases data set as Z in a d-dimensional space using a predetermined number of clusters. The next step is to link each of Z’s points with the cluster that has the closest prototype. The first step is accomplished, and an early group is finished when no points are outstanding. For the k prototypes to remain the most representative element of each cluster, we must recalculate their new values at this point. Consequently, it may

observe that the *k* prototypes gradually shift their office until there are no more modifications [6].

```
    def process_output(self, fetches, outputs) -> dict:
        # do a bit of postprocessing
        out_keys = ["label", "confidence"]
        results = {}
        # since we actually ran on a batch of size 1, index ou
t the items from the returned numpy arrays
        for i, (key, _) in enumerate(fetches):
            val = outputs[i].tolist()[0]
            if isinstance(val, bytes):
                val = val.decode()
            results[key] = val
        confs = results["Confidences"]
        labels = self.signature.get("classes").get("Label")
        output = [dict(zip(out_keys, group)) for group in zip(
labels, confs)]
        sorted_output = {"predictions": sorted(output, key=lam
bda k: k["confidence"], reverse=True)}
        return sorted_output
```

Figure 4.3 - K-Means Algorithm implementation

4.1.7. Image Classification

The architectures for the models we develop for image categorization are shown in this section. This layer runs a sliding window across the images, convolving the sub-image with filters at each step, resulting in a new volume with increasing depth. By a given factor, a pooling layer downsamples the volume along the spatial dimensions. The images in the testing dataset are also pre-processed and scaled to 225 × 225 x 3 during the testing phase [4]. Following that, the features from the images are retrieved and stored in the feature vector. The extracted feature vector is sent to the learned classifier, and the image is classified as healthy, hair loss, hot spots, ticks, rashes, swellings, lmps, and dry skin conditions and wounds.

```
# center crop image (you can substitute any other method to
 make a square image, such as just resizing or padding edge
s with 0)
        if width != height:
            square_size = min(width, height)
            left = (width - square_size) / 2
            top = (height - square_size) / 2
            right = (width + square_size) / 2
            bottom = (height + square_size) / 2
            # Crop the center of the image
            image = image.crop((left, top, right, bottom))
        # now the image is square, resize it to be the righ
t shape for the model input
        input_width, input_height = input_shape[1:3]
        if image.width != input_width or image.height != in
put_height:
            image = image.resize((input_width, input_height
))

        # make 0-1 float instead of 0-
255 int (that PIL Image loads by default)
        image = np.asarray(image) / 255.0
        # format input as model expects
        return image.astype(np.float32)
```

Figure 4.4 - Image classification implementation using CNN

4.1.8. Neural Networks

Layers will be created initially in order to build the network. Adding an input layer is the first step. Then there will be hidden layers added. There are many hidden layers in a deep neural network. A neural network with more hidden layers may be more effective. The output layers are now being added. The neural network must be built and fitted to the dogs' skin diseases training data after adding layers. Assume that the input matrix's reading starts at the top left corner of the image. The algorithm chooses a filter, which is a smaller matrix. The purpose of the filter is to multiply values with the original pixel values. All of these multiplications are added together.

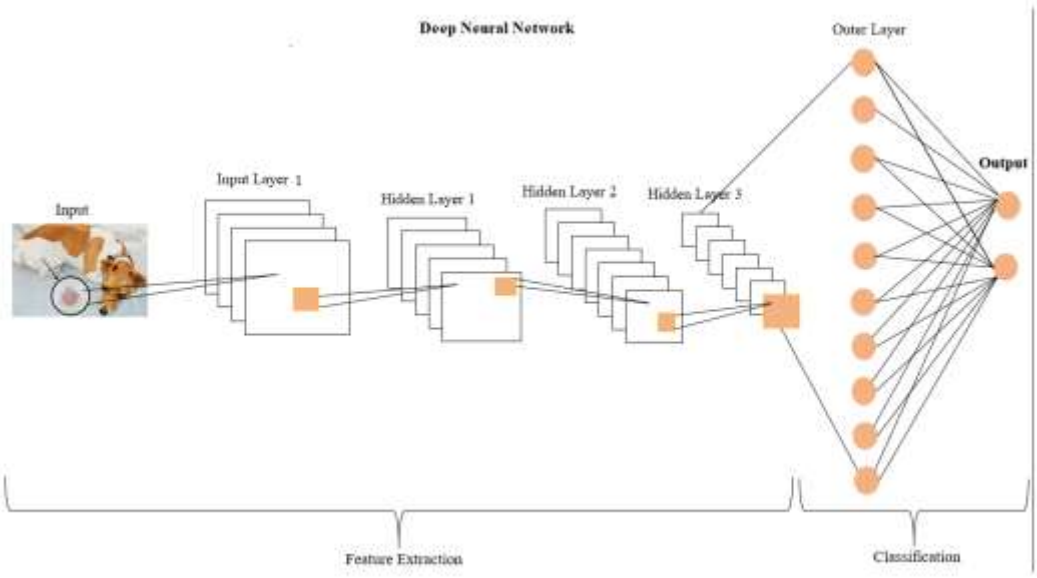


Figure 4.5 - Proposed System for CNN Architecture

Finally, only one number is acquired. Because the filter has only read the image in the top left corner, it moves 1 unit to the right, repeating the same process. A matrix is produced after running the filter across all positions. The pooling layer works with the image's width and height. In the previous convolution process, certain features had already been identified. When a detailed image is no longer required for processing, it is compressed into smaller images. As a result, the volume of the image is reduced. By changing these weights, the network's accuracy may be improved. The neural network's accuracy may be improved by repeating this process several times [14].

4.1.9. Implementation of chatbot

To build a chatbot from scratch, we must choose an NLP/ML framework and develop a neural network. NLP.js is a feasible option for this task since it focuses on Node with React implementation of chatbots [14]. Furthermore, NPM is the most convenient platform for hosting and running a React application's web server. First, We must build a corpus, which is a collection of question-and-answer entries and all of the necessary elements. Figure show the corpus in the quizeQuestion.js.

```
var quizQuestions = [
    {
        question: "Select your pet dog’s Breed ",
        answers: [
            {
                type: "0",
                content: "Labrador"
            },
            {
                type: "1",
                content: "German Shepherds"
            },
            {
                type: "2",
                content: "Golden Retrievers"
            },
            {
                type: "3",
                content: "Bulldog"
            },
            {
                type: "4",
                content: "Ridgeback"
            },
            {
                type: "5",
                content: "Rottweiler"
            },
            {
                type: "6",
                content: "Dalmatian"
            },
            {
                type: "7",
                content: "Beagle"
            },
            {
                type: "8",
                content: "Doberman"
            },
            {
                type: "9",
                content: "Pomeranian"
            },
            {
                type: "10",
```

```
                content: "Boxer"
            }
        ]
    },
    {
        question: "How old is your pet dog?",
        answers: [
            {
                type: "a",
                content: "Less than 1 month"
            },
            {
                type: "b",
                content: "1 month to 5 months"
            },
            {
                type: "c",
                content: "5 months to 1 year"
            },
            {
                type: "d",
                content: "Over 1 year"
            }
        ]
    },
    {
        question: "Severity of signs?",
        answers: [
            {
                type: "g",
                content: "Minor"
            },
            {
                type: "h",
                content: "Severe"
            }
        ]
    },
    {
        question: "How long have your pet dog had these symptoms for?",
        answers: [
            {
                type: "e",
                content: "Less than 5 days"
            },
```

```
{
                type: "f",
                content: "Above 5 days including 5 days"
            }
        ]
    },
];

export default quizQuestions;
```

Figure 4.6 - NLP.js Chatbot Implementation

The intents and their corresponding phrases that the chatbot will use to train itself can be found below. When the chatbot recognizes an intent, it will choose one of the available responses at random. It will also create a number or character for the potential responses to the chatbot's predictions. Because certain intents are complicated, they do not have matching responses. In addition, we must handle them using a callback function. The remaining code defines the unnamed function, the main loop, where all training and message exchange occurs. The diagram of following illustrates the chatbot responses structure's possibilities.

4.1.10. Implementation and the Deployment of the API

We have to build a REST API that can communicate with any service to take advantage of these models. I used the python-based flask framework for this. API has two endpoints in general. One is for detecting skin diseases, and the other is for getting a collection of questioners for a particular instance to predict treatments or veterinary-related areas. The sample request that the REST API allows is shown in the diagrams below.

```
@app.route('/disease', methods = ['GET', 'POST'])
def predict():
    data = {}

    filestr = request.files['file'].read()
    # convert string data to numpy array
    img = imgread(filestr)
    preprocess(filestr)

    prediction_inst = []
    list_Of_cf = []

    outputs = model_process_img(img)

    for item in outputs['predictions']:
        list_Of_cf.append(item['confidence'])

    for item in outputs['predictions']:
        if item['confidence'] == max(list_Of_cf):
            print(item['label'], max(list_Of_cf) * 100)
            prediction_inst.append(item['label'])

    temp_val = prediction_inst[0]
    print(temp_val)
    prediction_inst.clear()
    list_Of_cf.clear()
    data['detection'] = temp_val

    return jsonify(data)
```

Figure 4.7 - REST API implementation for Skin Diseases

4.2 Use case diagram of the system

The visual representation of a system's activity from the point of view, as specified by prescribed visual methods. They aid in the discovery and definition of functional requirements. The diagram demonstrates both skin diseases identification and chatbot conversation functional requirements.

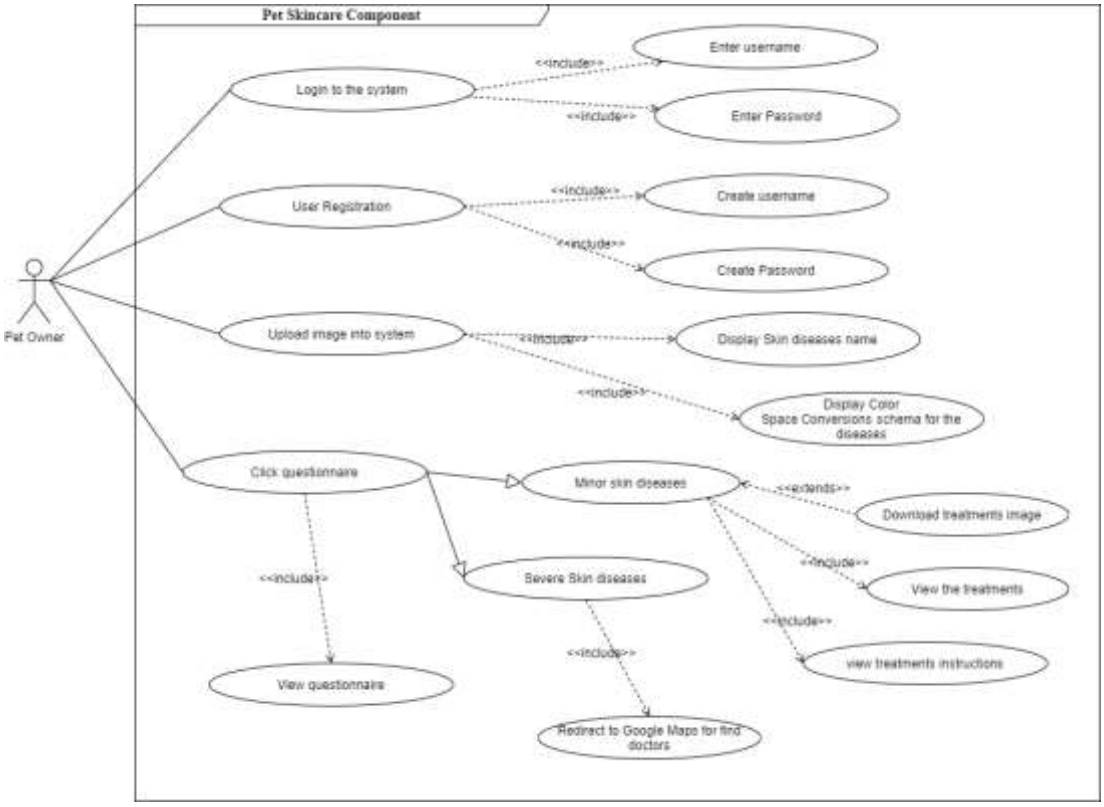


Figure 4.8 - Usecase Diagram for the pet skincare component

4.3 System Development Process

In this section of the proposed study, we detail how this section is designed and how it integrates with the main structure. We will use the Agile approach since this is a research and development project [14]. It allows continued iteration of development, Integrates and testing phases. The authors' solution would be focused on the system developed by the literature survey and the survey implemented, which will result in continual improvements since Scrum can test and adapt to evolving requirements.

Figure 4.9 - Agile Software Development Lifecycle

4.4 Gantt chart

The Gantt chart of the development process created according to the tentative deadlines is depicted in Figure 2.4.

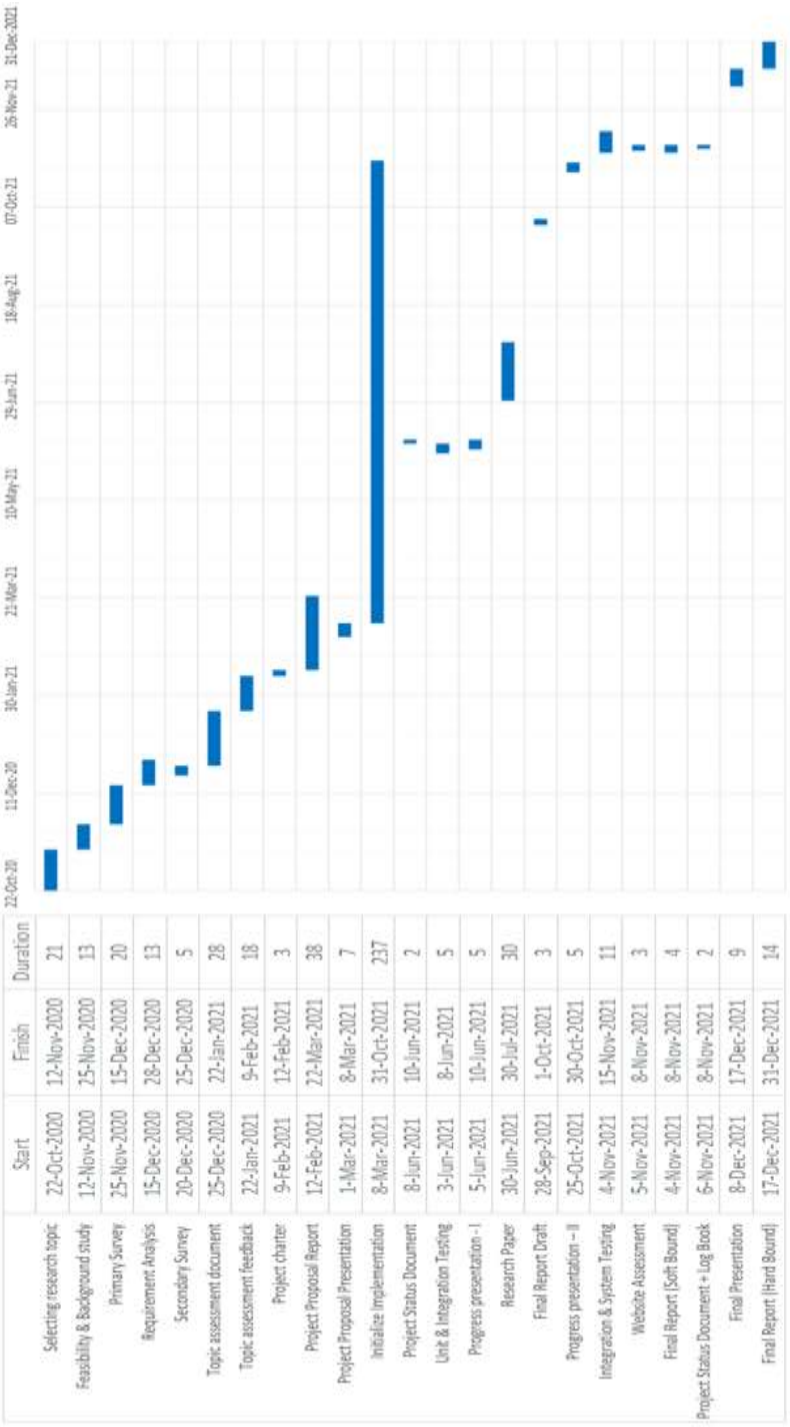


Figure 4.10 - Gantt Chart

4.5 Work Breakdown Structure

The work breakdown structure of the development process is shown in Figure

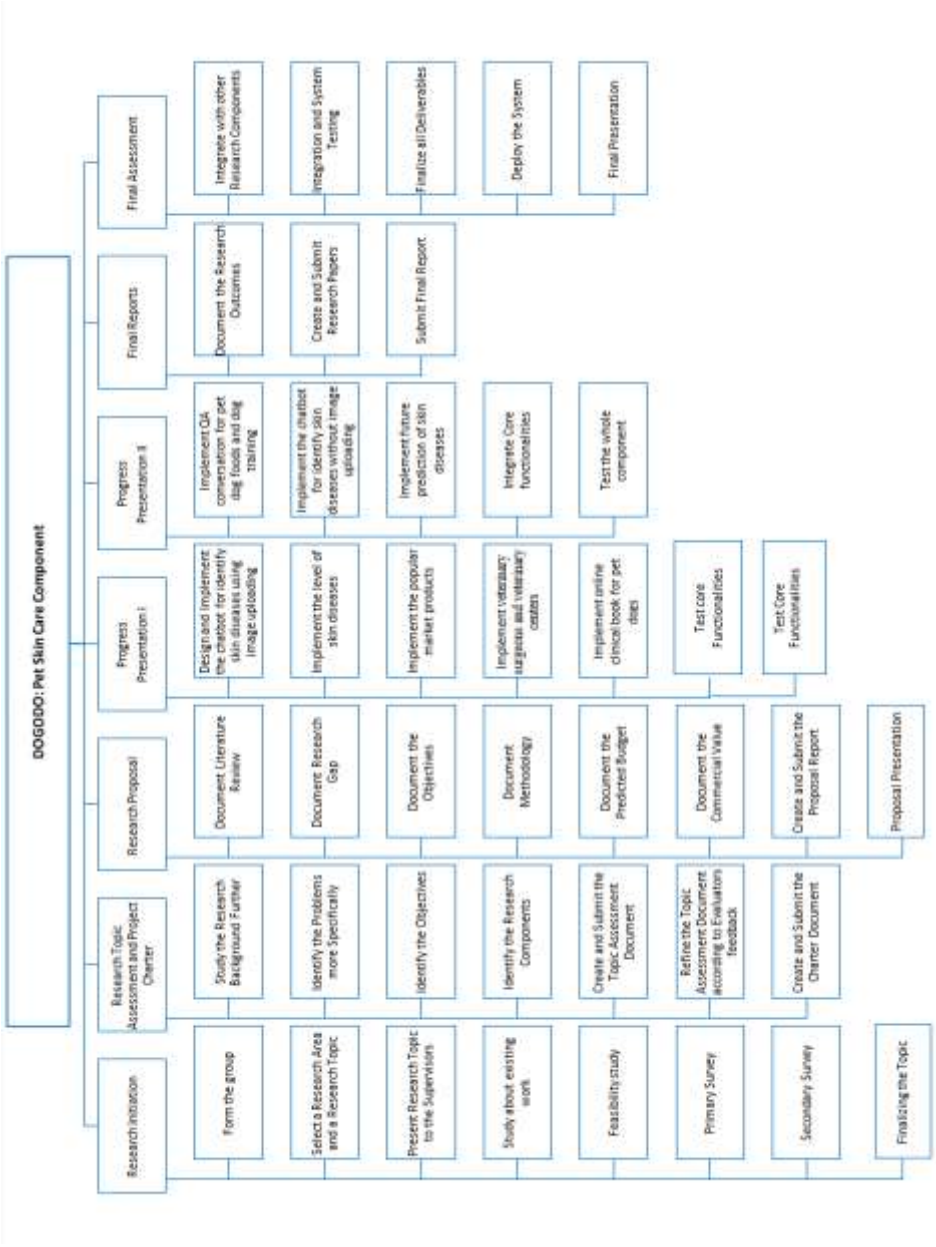


Figure 4.11 - Work Breakdown Structuer

4.6 Commercialization Aspect

4.6.1. Targeted Audience

The proposed solution will mainly target pet dogs with their owners. Dog owners/veterinarians /pet clinics will be the end-users of this system.

4.6.2. Benefits to the end-users

- The user can get a solution to their pet dog skin diseases in a short period
- Without going to a veterinary surgeon, identify skin diseases
- Improve the knowledge about pet dogs skin treatments and the instructions of the treatments via chatbot
- Get service from the system application 24-hours.

4.6.3. Advertising and Communication

Commercializing an all-purpose tool would be extra valuable for users in the target domain and can be considered a simple however rewarding innovative venture.

- The app is mainly promoted through social media
- Dog owners can usually share the device with supermarkets, grocery stores, and stores.
- A Facebook page and a YouTube channel will be established to advertise the mobile app in detail.

5 IMPLEMENTATION & TESTING, RESULTS & DISCUSSION

5.1 Testing Phase

- **Unit Testing**

Individual parts of source code are evaluated to see whether they are suitable for users using the unit testing technique. A unit is the smallest component of an application that can be tested.

- **Module Testing**

Each class, data, module, or component is tested as part of the module testing process. Module testing is carried out by a group member who is not the module's owner.

- **Intrigation Testing**

Individual software modules are integrated and evaluated in integration testing, which is a step of software testing. After module testing, but before validation testing, it happens. Integration testing combines unit-tested input modules into bigger aggregates, performs tests specified in an integration plan to those aggregates, and provides the integrated system suitable for system testing as an output. Individual members of the group will also carry out integration testing. Now is the time for all four members to contribute.

- **System Testing**

System testing of software or hardware involves evaluating a complete, integrated system with its stated criteria. System testing is a kind of black-box testing that does not need any understanding of the code's or logic's inner workings. The integrated system will be tested as a whole and changed if any errors are identified.

- **User Acceptance Test**

Bug-free apps will be created once the system testing is completed. As well, a target group will then do user acceptability testing. Selected parents and teachers will make up the target group. They will go through the system's features and provide recommendations for changes. The research team will evaluate the needs, and changes will be made.

Table 5.2.1 – Test Case for User Login

Test Scenario ID	Login-01	**Test Case ID**	Login 1-A
Test Case Description	Login -positive test case	**Test Priority**	High priority
Pre-Requisite	The user Login tho the system.	**Post-Requisite**	N/A

Test Execution Steps:

S. No	**Action**	**Inputs**	**Expected Output**	**Actual Output**	**Test Browser**	**Test Result**	**Test Comments**
1.	Browse the application	Cmd:npm start	Launch http://localhost:3000/	http://localhost:3000/	Google chrome	Pass	[Shashini 5.10 P.M. 10/10/2021]

							Launch successful
2.	The user enter the password and the username and click on the login button	Email id: abc@gmail.com Password :123	View user Login page	View user Login page	Google chrome	Pass	[Shashini 5.10 P.M. 10/10/2021] Launch successful

Table 5.2.2 – Test Case for View skin diseases results

Test Scenario ID	View skin diseases result - 02	**Test Case ID**	Login 1-A
Test Case Description	User view the results of skin diseases.	**Test Priority**	High priority
Pre-Requisite	User can view the result.	**Post-Requisite**	N/A

Test Execution Steps:

S. No	**Action**	**Inputs**	**Expected Output**	**Actual Output**	**Test Browser**	**Test Result**	**Test Comments**
1.	Browse the application	Cmd:npm start	Launch http://localhost:3000/	http://localhost:3000/	Google chrome	Pass	[Shashini 5.20 P.M. 10/10/2021] Launch

							succe ssful
2.	User click on the file uload button and select the image.	Image extention s: JPG/PNG / JPEG	User upload diseases image	User upload diseases image	Goog le chro me	Pas s	[Shas hini 5.20 P.M. 10/10/ 2021] succe ssful
3	Click on the the upload buton user can view the result.		User vew the result	User vew the result	Goog le chro me	Pas s	Shash ini 5.20 P.M. 10/10/ 2021] succe ssful

Table 5.2.3 – Test Case for View treatments for minor diseases

Test Scenario ID	View Treatments - 03	**Test Case ID**	Login 1-A
Test Case Descripti on	View skin diseases treatments via chatbot.	**Test Priority**	High priority
Pre-Requisite	User can start a conversation with the chatbot	**Post-Requisite**	N/A

Test Execution Steps:

S. No	**Action**	**Inputs**	**Expecte d Output**	**Actual Output**	**Test Brow ser**	**Te st Re sul t**	**Test Com ments**
1.	Browse the applicar ion	Cmd:np m start	Launch http://loc alhost:30 00/	http://local host:3000/	Goog le chro me	Pas s	[Shas hini 5.20 P.M.

							10/10/2021] Launch successful
2.	Start a conversation with chatbot click on the Questioner button.	Select the answers for the chatbot questions	Redirect to questions	View questions and user can response the questions	Google chrome	Pass	[Shashini 5.20 P.M. 10/10/2021] Launch successful
3	Select the minor skin diseases then it will redirect to the treatments page		View the skin diseases treatments	View the skin diseases treatments	Google chrome	Pass	[Shashini 5.20 P.M. 10/10/2021] Launch successful

Table 5.2.4 – Test Case for

Test Scenario ID	Find Doctors - 04	**Test Case ID**	Login 1-A
Test Case Description	User find doctors via chatbot.	**Test Priority**	High priority
Pre-Requisite	User can start a conversation with the chatbot	**Post-Requisite**	N/A
Test Execution Steps:			

S.No	Action	Inputs	Expected Output	Actual Output	Test Browser	Test Result	Test Comments
1.	Browse the application	Cmd:npm start	Launch http://localhost:3000/	http://localhost:3000/	Google chrome	Pass	[Shashini 5.20 P.M. 10/10/2021] Launch successful
2.	Start a conversation with chatbot click on the Questioner button.	Select the answers for the chatbot questions	Redirect to questions	View questions and user can response the questions	Google chrome	Pass	[Shashini 5.20 P.M. 10/10/2021] Launch successful
3	Select the severe skin diseases then it will redirect to the Google Maps.		View the Google Maps	View the Google Maps	Google chrome	Pass	[Shashini 5.20 P.M. 10/10/2021] Launch successful

5.2 Results & Research Findings

5.3.1 Results

This method forced the dogs' skin problems to be identified. Traditional systems of recognizing skin diseases have some drawbacks. Pet owners can usually accurately recognize skin diseases using this system. I used a large number of skin images (both infected and healthy dogs) to train a model using a machine learning and deep learning architecture since the goal of this study is to control the spread with the function of identifying the diseases. Furthermore, at the end of the training, the model will identify the location of diseased areas in any skin disease given an image of the disease as input and improve its efficiency.

The model is trained using the training dataset. The model learns its weights and biases in the case of neural networks. After each set of predictions, the model evaluates itself using the validation dataset. It improves the model's hyperparameter optimization. After the model has been appropriately trained, the test dataset is used to evaluate it. Because the whole dataset cannot continuously be fed into the neural network, it has to be split into parts or batches. The batch size determines how many training samples are included in a single batch.

With a web-based application, the system was implemented in a mobile-first approach. It can be used on both Android, Tablet, and computer devices. With some modifications, this model is built on TensorFlow's CNN. Three convolutional layers are followed by a max-pooling layer and two dropout layers with a 0.1 dropout rate to avoid overfitting. Then there are two dense layers, each having 255 and 9 units (9 is the number of classes for classification). There is a dropout layer with a dropout rate of 0.1 between

the two dense layers. The batch size is 32, and there are 100 epochs, and the learning rate is 0.01.

The classification metrics are generated first by running the TensorFlow framework's model_process_img.py script, which runs through our trained model and saves the results in a saved_model record file. The accuracy of the training and testing dataset is relatively high: 92.3% and 91%, respectively. Due to a lack of time and hardware resources, and this model is the final model. As you can be observed, figure 5.1 shows the compared to the unshifted (top) plot, shifting the training loss values a half epoch to the left (bottom) makes the training and validation curves considerably more comparable.

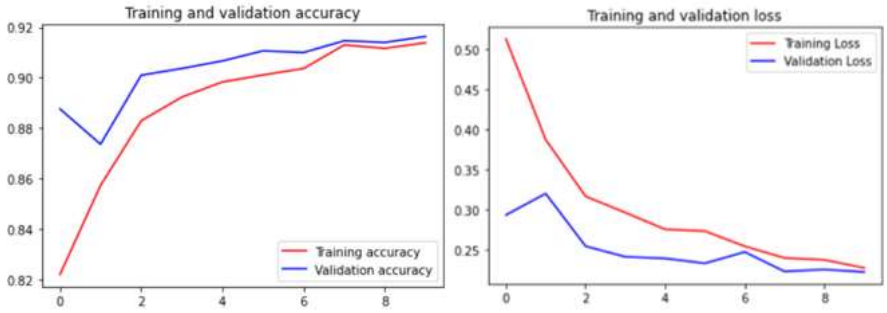


Figure 5.1 - Train/Validation Acurracy and Loss plot diagram

5.3.2 Research Findings

When we consider the research findings, Figure, Figure, Figure, Figure, Figure, Figure, Figure, Figure, and Figure illustrate the findings. The identification of the cropped data is successful in all cases, according to the findings.

- **Hair Loss skin disease**

Figure 5.2 shown the accuracy of hair loss skin disease. It is shown that the accuracy of the image classification for hair loss diseases was 100.00% when.It was simulated and compared to other types of skin disease. As well as shown in figure 5.3, Color Space Conversions (CSC) schemes result from the hair loss disease.

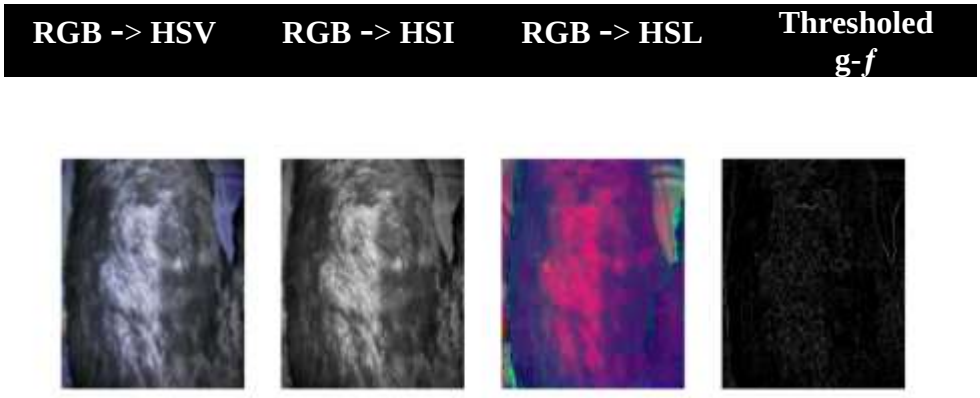


Figure 5.2 - Color Space Conversions for Hair Loss

Figure 5.3 - Accuraacy for the Hiar Loss Skin Disease

Hot Spot skin disease

The accuracy of hot spots skin disease is demonstrated in Figure 5.4. When stimulated and compared to other skin diseases, the picture classification accuracy for hot spots diseases was 100.00%. CSC schemes are caused by the hot spots diseases, as illustrated in figure 5.5.

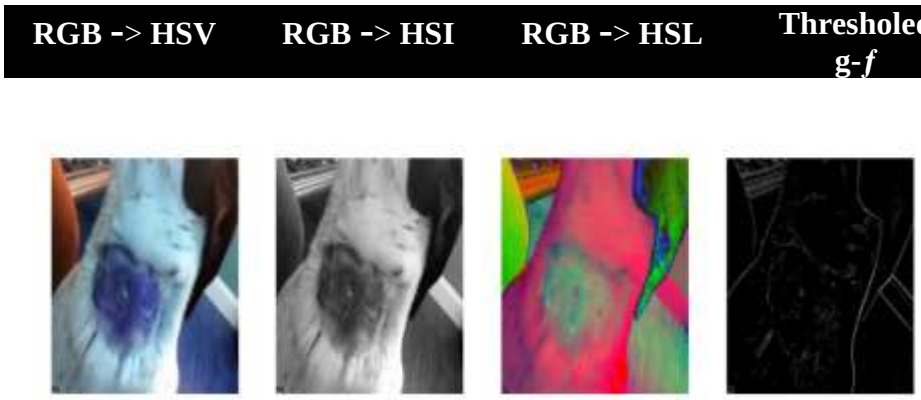


Figure 5.4 - Color Space Conversions for Hot Spot

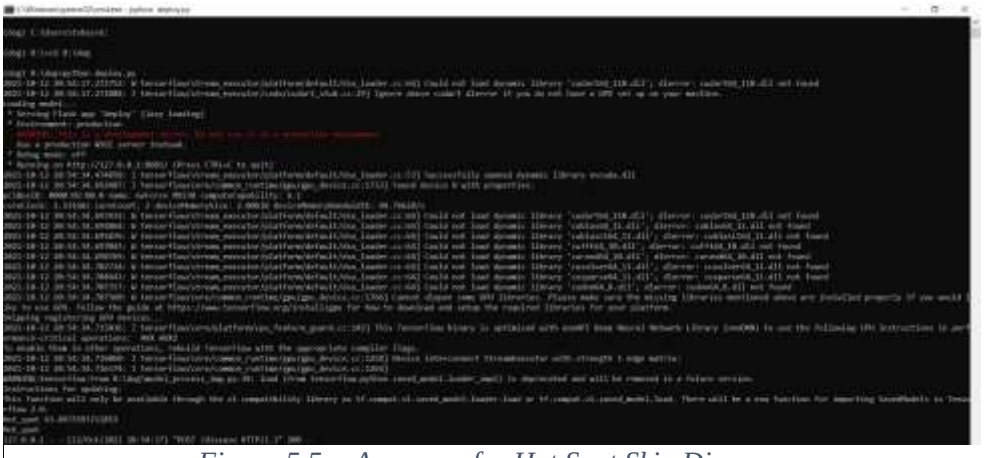


Figure 5.5 - Accuracy for Hot Spot Skin Diseases

Lumps skin disease

Figure 5.6 shown the accuracy of Lumps skin disease. It is shown that the accuracy of the image classification for Lumps diseases was 100.00% when it was simulated and compared to other types of skin disease. As well as shown in figure 5.7, CSC schemes result from the Lumps disease.

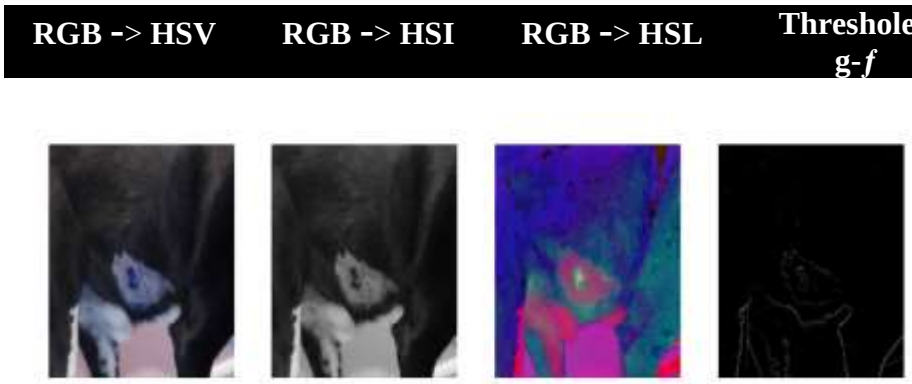


Figure 5.6 - Color Space Conversions for Lumps

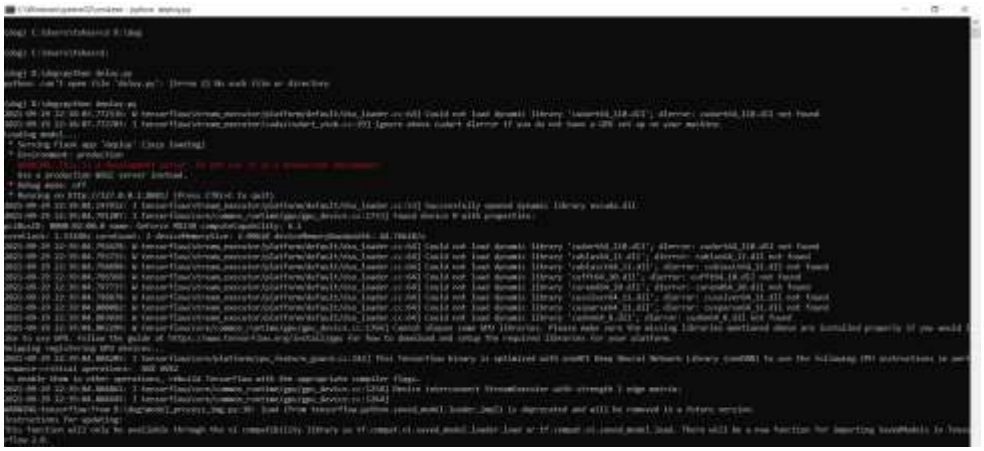


Figure 5.7 - Accuracy for Lumps

Rashes skin disease

The accuracy of rashes skin condition is demonstrated in Fig 5.8. When stimulated and compared to other skin diseases, the image classification accuracy for rashes diseases was 100.00%. Color CSC schemes arise as a result of the rashes disease, as shown in figure 5.9.

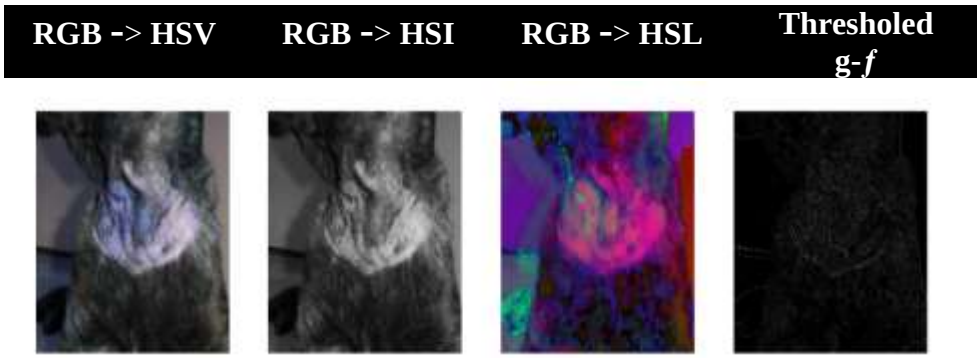


Figure 5.8 - Color Space Conversions for Rashes

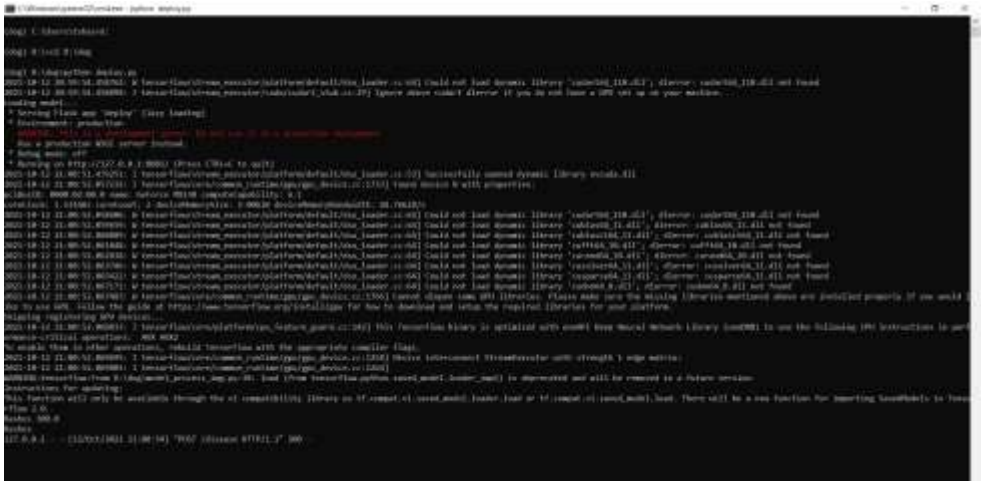


Figure 5.9 - Accuracy for Rashes

Swelling skin disease

Figure 5.10 shows the accuracy of swelling skin conditions. When stimulated and compared to other types of skin disease, the image classification accuracy for swelling diseases was 100.00 per cent. CSC schemes are caused by swelling disease, as illustrated in figure 5.11.

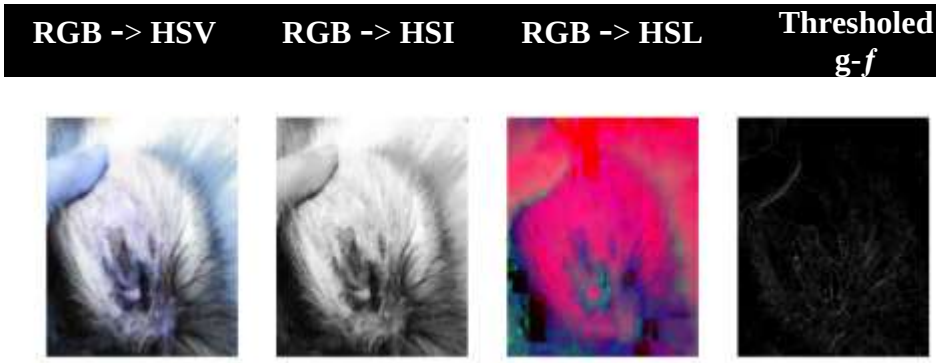


Figure 5.10 - Color Space Conversions for Swelling

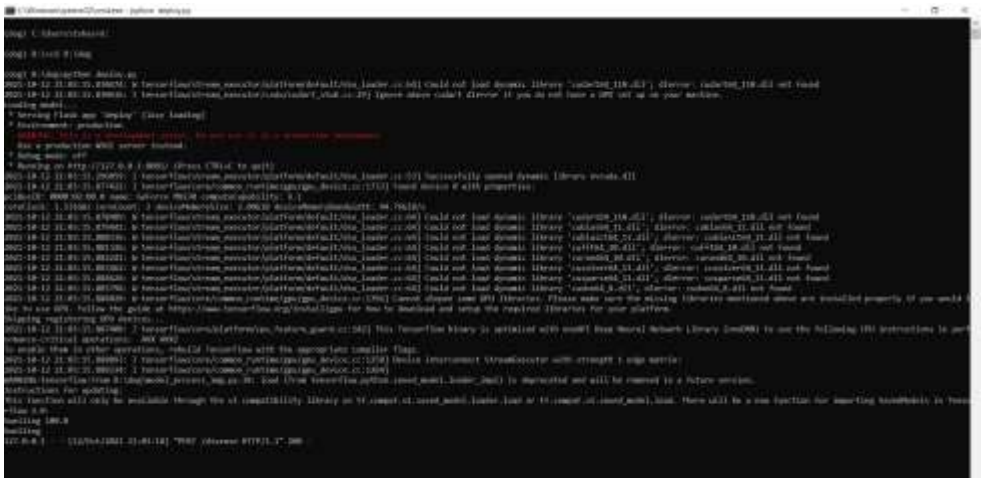


Figure 5.11 - Accuracy for Swelling

Ticks skin disease

Figure 5.12 shown the accuracy of ticks skin disease. It is shown that the accuracy of the image classification for ticks diseases was 100.00% when it was simulated and compared to other types of skin disease. As well as shown in figure 5.13, CSC schemes result from the ticks disease.

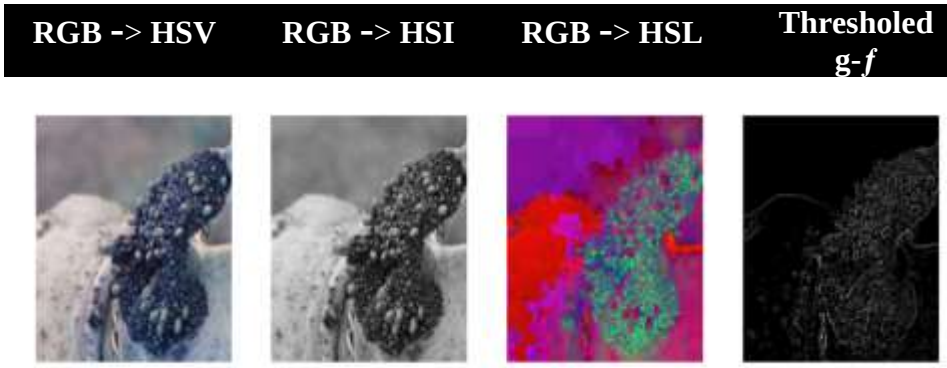


Figure 5.12 - Color Space Conversions for Ticks

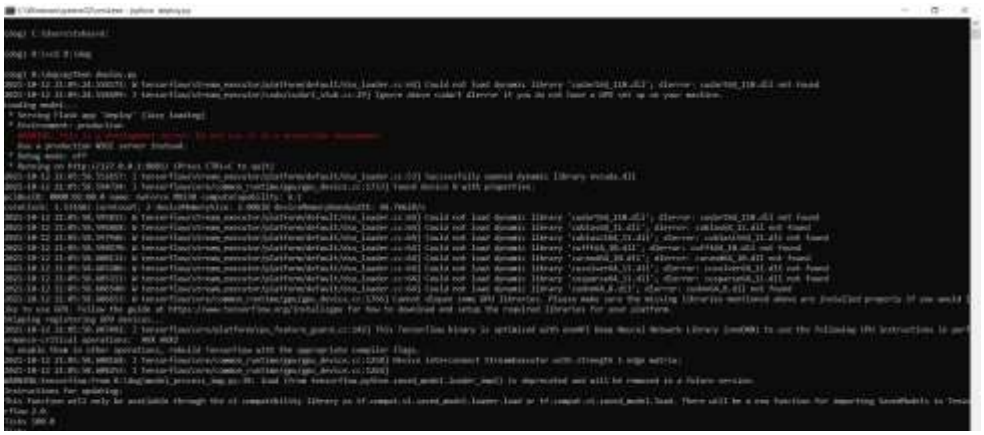


Figure 5.13 - Accuracy for Ticks

Pet dogs' Wounds

The accuracy of pet dogs wounds is illustrated in Figure 5.14. When stimulated and compared to other types of skin disease, the image classification accuracy for the wound was 100.00%. CSC schemes are caused by wounds, as shown in figure 5.15.

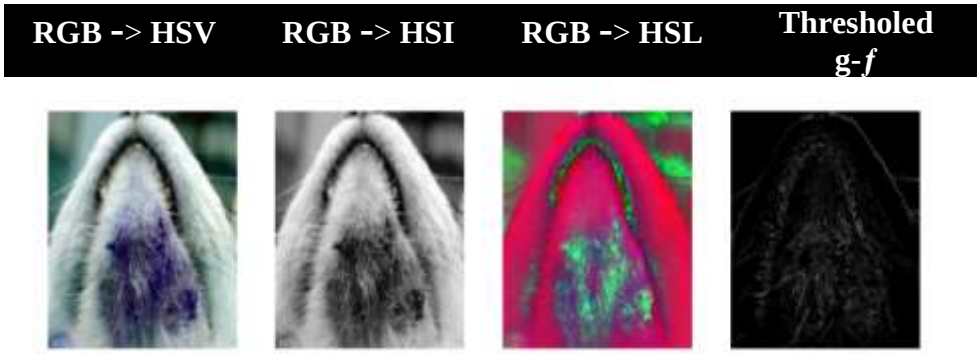


Figure 5.14 - Color Space Conversions for pet dog's wounds

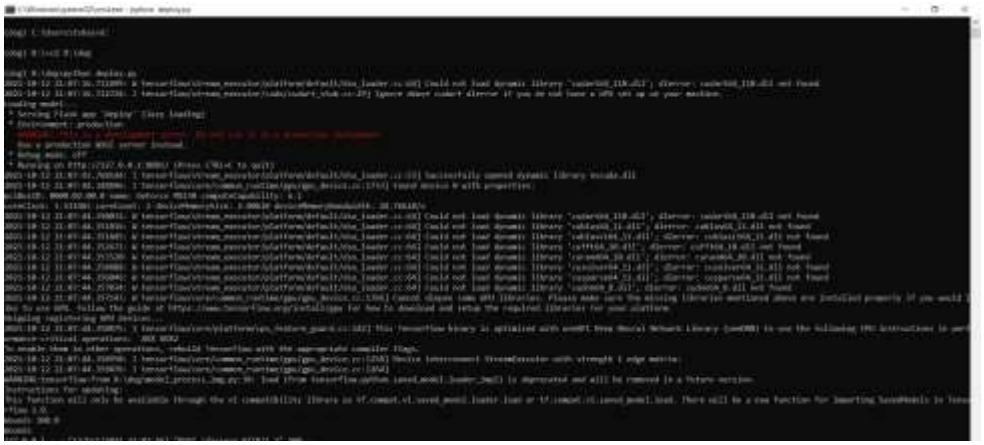


Figure 5.15 - Accuracy for pet dog's wounds

Healthy Dogs

Figure 5.16 shown the accuracy of non-infected pet dogs. It is shown that the accuracy of the image classification for healthy dogs was 100.00% when it was simulated and compared to other types of skin disease. As well as shown in figure 5.17, CSC schemes result from healthy dogs.

Figure 5.16 - Color Space Conversions for Healthy Dogs

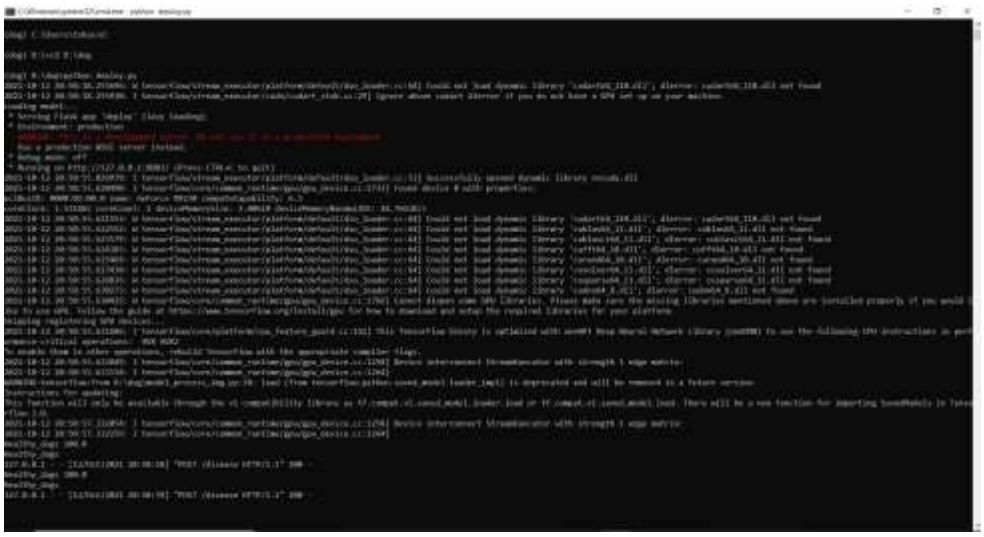


Figure 5.17 - Accuracy for healthy dogs

5.3 Discussion

The result of this study will be determined by the goals that must be achieved. Apart from that, certain parameters had a role in the convolutional neural network's image classification accuracy. The first result of this study was placed to the test by identifying the different types of skin diseases that affect dogs. It can be observed that in terms of implementing the system of image classification by using Multiclass Classification, all eight (8) types of pet dogs' skin diseases demonstrated up to 54% accuracy. The first result of the training accuracy keeps improving, but the validation accuracy reaches a plateau rather quickly. When a model fails to reflect the underlying trend of the data or does not intuitively match the data well enough, it is called to be underfitting. This happened due to the lack of a proper dataset.

```
[ ] # Flattenning the output of the Inception
    x = Flatten()(resnet.output)

[ ] # Multi calsss classification and concatanate with previous Inception
    prediction = Dense(len(folders), activation='softmax')(x)

    # create a model object
    model = Model(inputs=resnet.input, outputs=prediction)
```

Figure 5.18 - Multiclass Classificarion Algorithm

```
[ ] # plot the loss
    plt.plot(r.history['loss'], label='train loss')
    plt.plot(r.history['val_loss'], label='val loss')
    plt.legend()
    plt.show()
    plt.savefig('LossVal_loss')
```

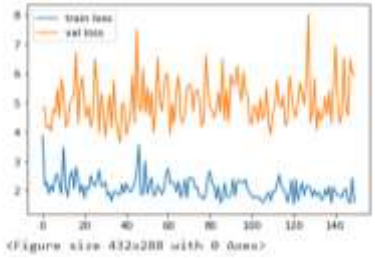


Figure 5.19 - Accuracy of training and valid dataset using multiclass classification

The proposed system was tested using an image of Hot_Spot 3.jpg, which was not used in the training model. The images were also not in the Hot_Spot category. instead, they were in the Hair_loss category. Following the categorization, there were some errors. The errors indicate that the images cannot be identified by the systems since they were not trained such that the model trained can identify it as the skin disease identified as Hair_loss.

Therefore, Used the K-means algorithm and the CNN for the next training phase. As well as using RGB, HSV, HSL, and canny morphological edge with thresholded for feature extraction. Lastly, the results of the graph showed some changes when one of the parameters was changed. The size of the model was set into two different sizes, and it affected the systems. The model's size became prominent, and the training session took a considerable time to be completed. However, the percentage of 92.3% accuracy with high compared to the large size of the training model of the skin diseases dataset.

When talking about the chatbot, it depends on the skin diseases image. After identifying the diseases, the pet owner can start a conversation with the chatbot. Consider the severity of the skin diseases, and the chatbot will suggest the treatments or notify the owner with veterinary attention and the veterinary centers show within Google Maps.

5.4 Summary of Each Student's Contribution

Table 5.5.1 – Individual Contribution

Registration Number	Name	Functions
IT18001730	T.S. Chethana Fernando	• Implement the pet dogs' skin diseases identification system. • Implement the level of skin diseases and compare the other diseases with the accuracy of probability of the diseases. • Implementation of colour schema pattern for the uploaded skin diseases. • Implement a chatbot system. • Implementation of chatbot API • Implementation of pet dogs treatments for minor skin diseases. • Implement a suggestion for finding the veterinarians (For severe issues) via a chatbot system.

6 CONCLUSION

The study's objectives were achieved since the developed application can capture localized skin diseases using the phone's back camera. According to the findings, the suggested application allows for detecting and treating dog skin diseases at home. As a result, the next phase will be to use image processing methods to identify different skin diseases in the same phase. A total of 2 639 images were used in this research. There are 924 images for training and 715 images for skin diseases in the test dataset.

It has three objectives that have been achieved throughout this research. It has three objectives that have been achieved during this study. The objectives are linked to the conclusions since they may determine whether or not all objectives are achieved. It may be stated that all of the results obtained were quite impressive. This study focuses on the convolutional neural network, which is particularly useful in image categorization technologies. The CNN method was studied in more detail, beginning with the assembly, training model, and classification of images into categories. In CNN, the function of epochs was able to control accuracy while also preventing problems from underfitting. Some problems were encountered during the study and were able to be resolve by increasing the image dataset, changing the hyperparameters, Using RBG, HSL, HSI, Morphological edge threshold for feature extraction, and using K-means and the CNN algorithm [7], [9], [10].

Deep learning implementation utilizing the TensorFlow framework also had good results, as it was able to stimulate, train, and classify eight different types of skin diseases and healthy dogs with up to 92.3% accuracy. Eventually, Python was chosen as the programming language throughout this study since it is compatible with the TensorFlow framework, which allows for the entire development of the proposed system's back end in Python.

The observations and tests suggested the RGB−HSL, RGB−HSI, RGB−HSV and threshold g-f color space because it used minimal resources and less power than other color systems. The new approach segments the component with the lowest intensity from the given image first at each iteration. The recursive process continues until the image includes only one object (the darkest). The RGB to HSL, HIS, HSV segments are modified by the conversion formula is shown by the following equation [5].

$$R' = \boldsymbol{R}/_{\boldsymbol{255}} \;;\; G' = \boldsymbol{G}/_{\boldsymbol{255}} \;;\; B' = \boldsymbol{B}/_{\boldsymbol{255}}$$

$$\boldsymbol{C_{max}} = MAX(R', G', B')$$

$$\boldsymbol{C_{min}} = MIN(R', G', B')$$

$$\blacktriangle = \boldsymbol{C_{max}} - \boldsymbol{C_{min}}$$

Figure 6.1 - RGB colour equation

Chatbots that use patterns have a questionable level of intelligence. It has a high learning rate, which means it takes up detail immediately. The chatbot aims to have a smart, accurate, and real-time conversation with the owners of pet dogs. The user of this application can specify their pet dog age, breed, the severity of diseases, disease duration, symptoms to the chatbot, and in turn, the chatbot will specify the health measures to be taken by symptoms. The chatbot instance can provide information about disease and treatment to the user. After analyzing the pet dog's age, breed, the severity of diseases, disease duration, and symptoms of the pet dog, it finally predicts the disease to the user and provides a link where details about the treatment are visible. Owners of pet dogs may upload an image to an app or an online platform and get a prediction without paying any costs or limiting their availability via chatbot. After diagnosing a pet dog's skin disease, an intelligent medical

chatbot helps dog owners provide the appropriate skin disease treatment. People rarely visit veterinary hospitals or centers for check-ups pet dogs in their busy lives. In such cases, the chatbot is helpful because it can provide treatment help with only one button click. One of the main benefits of the chatbot is that it does not need the help of a veterinarian to provide proper health measures to the users.

Furthermore, the mobile-first approach used in this project's web application will be a helpful tool for pet dog owners, especially those who live in areas with limited infrastructure and veterinary services. The pet owners identify the disease, which minimizes the risks, and there is no need to contact a veterinarian in minor cases. Dog owners will benefit from using Google Maps to locate a veterinarian specialist for serious skin disease issues.

REFERENCES

[1] Reand Michael M. Mellores, Jessa Carrisse D. Salvoza, Shantal O. Flores, Jan Carlo T. Arroyo, Allemar Jhone P. Delima, "Android-based Application Utilizing Image Processing with Artificial Neural Network for Detecting Ringworm and Yeast Infections for Dogs Using Neuroph Framework," [Accessed: 24- Mar-2021].

[2] "Dog Skin Conditions | Causes, Symptoms and Treatment," [Online]. Available: https://www.vets-now.com/pet-care-advice/dog-skin-conditions/. [Accessed 26 Feb 2021].

[3] Munirah M. Y., Suriawati S., and Teresa P. P, "Design and Development of Online Dog Diseases," *International Journal of Information and Education Technology,* vol. 6, 2016.

[4] J. Quintanilla-Dominguez, B. Ojeda-Magaña, M. Cortina-Januchs, R. Ruelas, A. Vega-Corona and D. Andina, "Image segmentation by fuzzy and possibilistic clustering algorithms for the identification of microcalcifications," vol. 18, no. 3, [Accessed: 26- Jul- 2021].

[5] G.Saravanan, G.Yamuna, and S.Nandhini, "Real Time Implementation of RGB to HSV/HSI/HSL and Its Reverse Color Space Models," in *2016 International Conference on Communication and Signal Processing (ICCSP)*, [Accessed: 26- Jul- 2021].

[6] Meenakshi S R, Arpitha B Mahajanakatti, Shivakumara Bheemanaik, "Morphological Image Processing Approach Using K-Means Clustering for Detection of Tumor in Brain," *International Journal of Science and Research (IJSR),* [Accessed: 26- Jul- 2021].

[7] R. Sumithra, M. Suhil and D. Guru, "Segmentation and Classification of Skin Lesions for Disease Diagnosis," in *International Conference on Advanced Computing Technologies and Applications (ICACTA*, [Accessed: 24- Mar- 2021].

[8] N. H. I. A. H. A. R. N. A. S. I. A. Mohd Azlan Abu, "A study on Image Classification based on Deep Learning and Tensorflow," in *International Journal of Engineering Research and Technology*, [Accessed: 24- Mar- 2021].

[9] Karan Chauhan, Shrwan Ram, "Image Classification with Deep Learning and Comparison between Different Convolutional Neural Network Structures using Tensorflow and Keras," *International Journal of Advance Engineering and Research Development*, [Accessed: 24- Mar- 2021].

[10] GWENDAL DANIEL, JORDI CABOT, LAURENT DERUELLE, MUSTAPHA DERRAS, "A Multimodal Low-Code Chatbot Development Framework," *reparation of Papers for IEEE TRANSACTIONS and JOURNALS*, [Accessed: 26- Jul- 2021].

[11] "Kommunicate," [Online]. Available: https://www.kommunicate.io/blog/nlp-libraries-node-javascript/. [Accessed 12 Oct 2021].

[12] Victoria Oguntosin, Ayobami Olomo, "Development of an E-Commerce Chatbot for a University Shopping Mall," *Hindawi Applied Computational Intelligence and Computing*, p. 14 pages, [Accessed: 24- Mar -2021].

[13] R. Sumithra, M. Suhil and D. Guru, "Segmentation and Classification of Skin Lesions for Disease Diagnosis," *International Conference on Advanced Computing Technologies and Applications (ICACTA2015)*, [Accessed: 26- Jul- 2021].

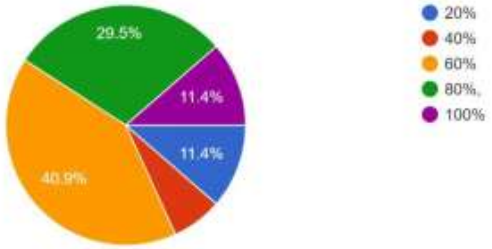
[14] Anooja Antony, Arun Ramesh, Asha Sojan, Betsy Mathews, Mrs. Tessy Annie Varghese, "Skin Cancer Detection Using Artificial Neural Networking," *INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING*, vol. 4, no. 4, [Accessed: 26- Jul- 2021].

[15] Abderrahim Bourouis, AliZerdazi, MohammedFeham, AbdelhamidBouchachia, "M-Health: Skin Disease Analysis System Using Smartphone's Camera," *The 8th International Symposium on Intelligent Systems Techniques for Ad hoc and Wireless Sensor Networks (IST-AWSN)*, vol. 19, pp. 1116-1120, [Accessed: 26- Jul- 2021].

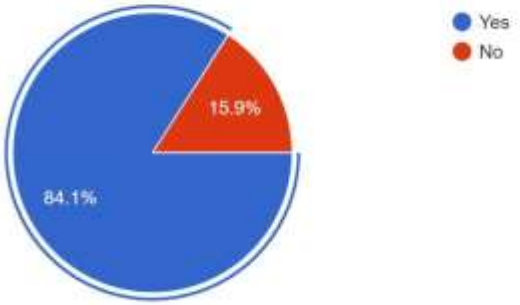
APPENDICES

Appendices - A

1. How well do think you understand your dog?
44 responses

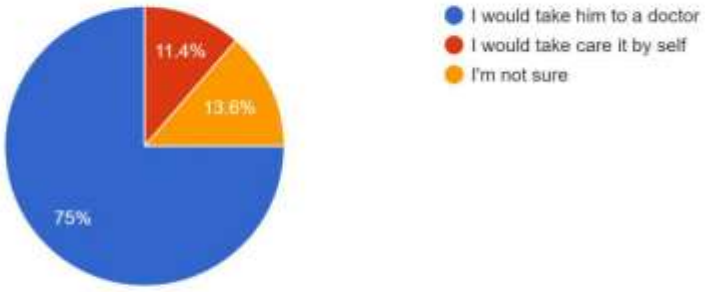


2. Can you understand if the dog is suffering from an illness?
44 responses



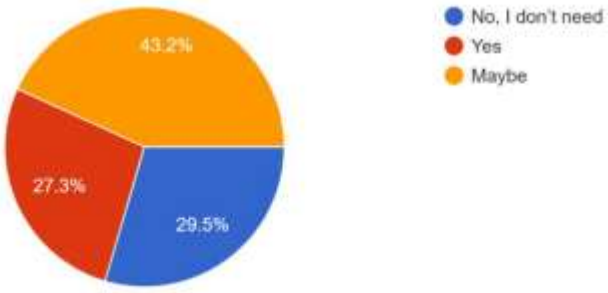
3. If you understand that the dog is suffering what would you do?

44 responses



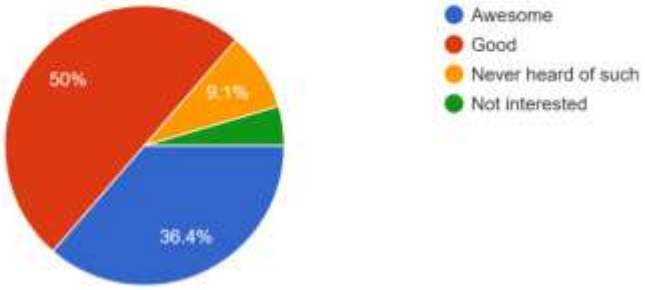
4. Would you like to have a device that would help the understand the dog better?

44 responses



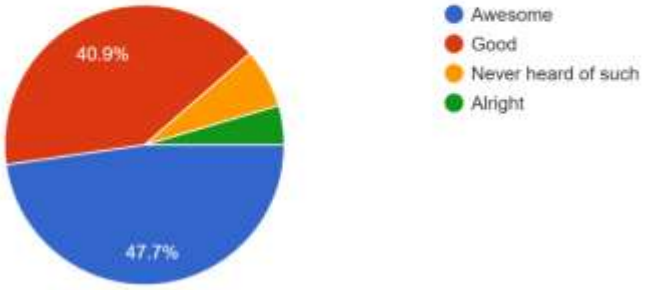
5. Would it be easy for you to have a device which can generate your Pet dog's internal health such as (temperature/footsteps (calories)/respiratory levels)?

44 responses



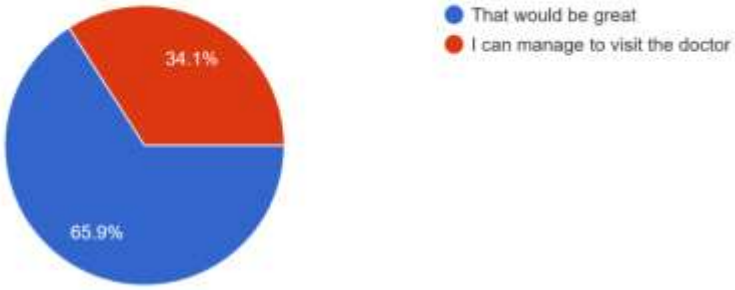
6. What do you think about a system which can predicts exercises and diet plans by tracking dogs foot steps and heart rate levels for the future health benefits?

44 responses



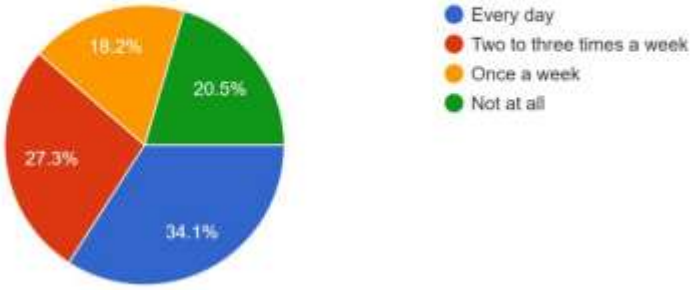
7. How do you rate the process of taking your dog to a doctor to a small issue which can be sorted out by a drug store with correct details?

44 responses



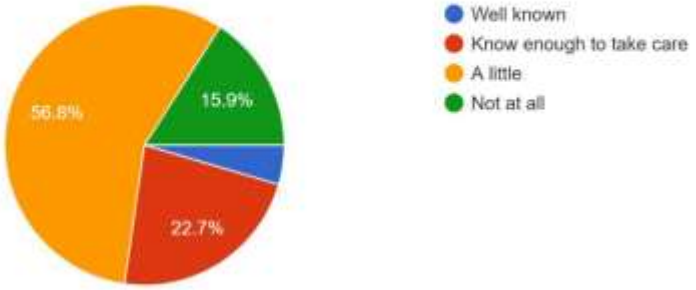
8. How often do you take your dog to a walk?

44 responses



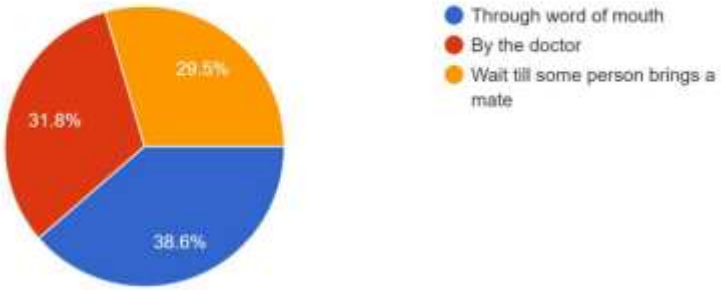
9. How do you know what result will come after a cross dog mix breed?

44 responses



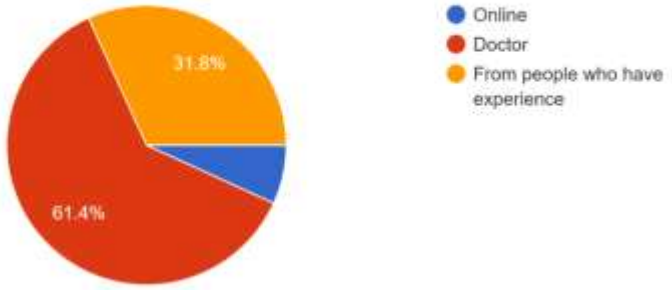
10. How do you normally find a partner for your pet?

44 responses



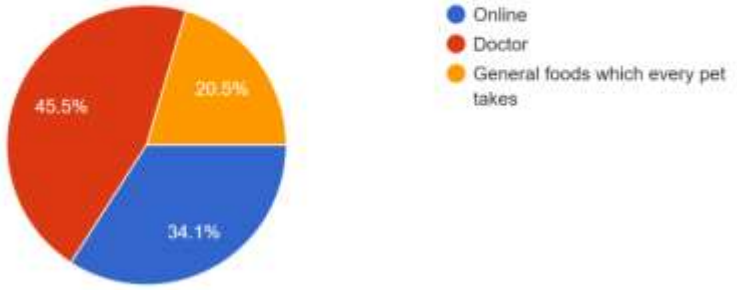
11. How to find necessary food recommendations for your pet?

44 responses



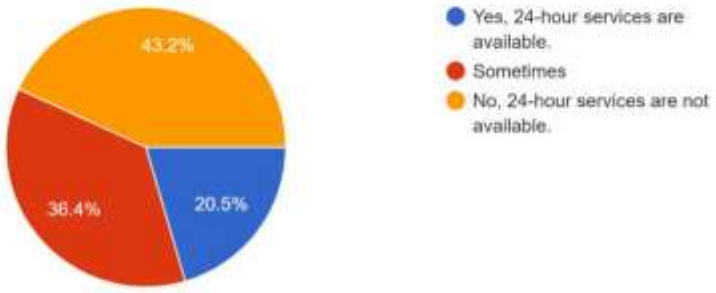
12. If you are not going to channel a doctor for your pet how do you find medical treatment for your newborn pet?

44 responses



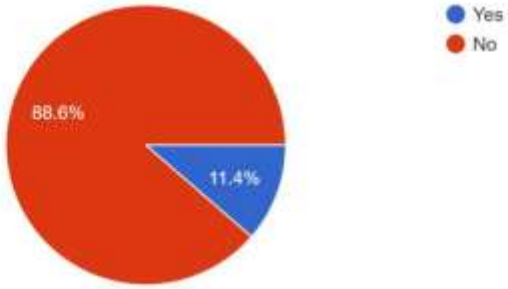
13. In your area, are veterinary services available 24 hours a day?

44 responses



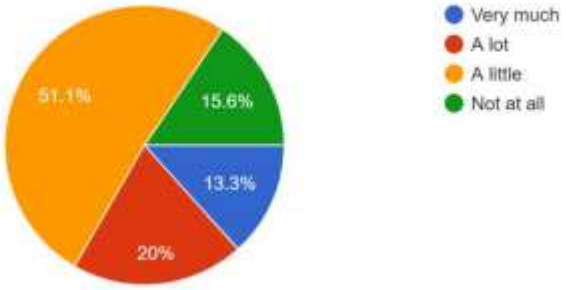
14. Have you ever used a mobile app to identify your pet dog's skin diseases?

44 responses



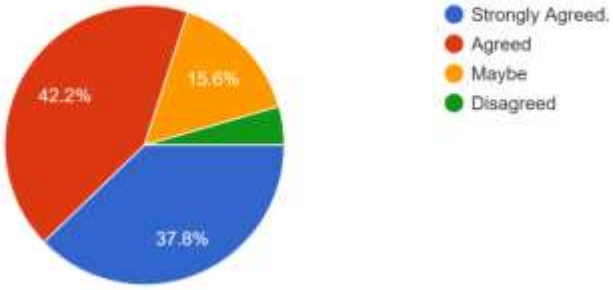
15. Do you pay attention to your dog's skin 24 hours a day?

45 responses



16. If there is a mobile application system for identifying small skin diseases without meeting a doctor, it helps identify skin di.... Do you agree with those mentioned above?

45 responses



Appendices - B

- Implementation of Skin diseases Identification

```
import React from "react";
import { db, auth } from './firebase';

import bgimg from './images/bghm.png';
import dog from './images/dog.jpg';
import doggr from './images/diseases/dogg.jpg';
import doghsv from './images/diseases/doghsv.jpg';
import dogedge from './images/diseases/dogedge.jpg';
import gaus from './images/diseases/gaus.jpg';
import thumb from './images/thumb.png';
import ques from './images/ques.svg';
import './App.css';
import Container from 'react-bootstrap/Container'
import Row from 'react-bootstrap/Row'
import Col from 'react-bootstrap/Col'
import Image from 'react-bootstrap/Image'
import Form from 'react-bootstrap/Form'
import Button from 'react-bootstrap/Button'
import "./styles.css";
import App from './App';
import ReactDOM from 'react-dom';
import Navbar from 'react-bootstrap/Navbar'
import Card from 'react-bootstrap/Card'
import CardGroup from 'react-bootstrap/CardGroup'
import Nav from 'react-bootstrap/Nav'
import DropzoneComponent from './DropzoneComponent';
import axios from 'axios'
import Alert from 'react-bootstrap/Alert'
import Home from "./Home";
import Location from './Location';

class Disease extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      imageURL: '',
      message: '',
      // messagestatus: false,
```

```
  };

  this.handleUploadImage = this.handleUploadImage.bind(this);
}


onImageChange = event => {
  if (event.target.files && event.target.files[0]) {
    let img = event.target.files[0];
    this.setState({
      imageURL: URL.createObjectURL(img),
      // messagestatus: true,
    });
  }
};


handleUploadImage(ev) {
  ev.preventDefault();

  const data = new FormData();
  data.append('file', this.uploadInput.files[0]);
  // data.append('filename', this.fileName.value);

  fetch('/disease', {
    method: 'POST',
    body: data,
  }).then(response => response.json()).then(response => {

    console.log(response['detection']);
    // alert(response['detection']);
    var msg = response['detection'];

    db.collection('disease').doc('disease').set({
      disease: msg
    })
    db.collection('symptoms').doc('symptoms').update({
      disease: msg
   });

  });
}
```

```
  backfunc() {
    ReactDOM.render(<Home />, document.getElementById('root'));
  }

  qfunc() {
    ReactDOM.render(<Location />,
document.getElementById('root'));
  }

  alertfunc() {
    const dis = []
    db.collection('disease').get().then(
      snapshot => {

        snapshot.forEach(doc => {
          const data = doc.data()
          dis.push(data)
          // console.log(dis[0]['disease'])
          this.setState({
            message: dis[0]['disease']
          })
        })
      }
    )

    // this.setState({
    //   message: 'a'
    // })

    // console.log(this.state.message)

    // if (this.state.messagestatus) {
    //   console.log(this.state.message)
    // }
    // console.log(this.state.message)

  }

  render() {
    return (
      <div>

        <div>
```

```
          <Navbar bg="dark" variant="dark">
            <Container>
              <Navbar.Brand >Doggo care</Navbar.Brand>
              <Navbar.Toggle />
              {/* <Navbar.Collapse className="justify-content-
end">
                <Navbar.Text>
                  Log out
                </Navbar.Text>
              </Navbar.Collapse> */}
            </Container>
          </Navbar>
        </div>
        <div>
          <h1 className='headerdisease'>Select you image</h1>
        </div>
        <Container>
          <Row>
            <Col>
              <form onSubmit={this.handleUploadImage}>
                <div>
                  <input ref={(ref) => { this.uploadInput =
ref; }} type="file" onChange={this.onImageChange}
className='inputdisease' />
                </div>
                <br />
                <div className='submitdisease'>
                  <button
className='buttondisease'>Upload</button>
                </div>
                <div className='submitdisease'>
                <img className='preview'
src={this.state.imageURL} alt="Preview of the selected dog" />
                </div>
              </form>
            </Col>
            <Col className='processedcol'>
              <div>
                <img src={gaus} className='processed' />
              </div>
            </Col>
            <Col className='processedcol'>
              <div>
```

```
                <img src={doggr} className='processed' />
              </div>
            </Col>
            <Col className='processedcol'>
              <div>
                <img src={doghsv} className='processed' />
              </div>
            </Col>
            <Col className='processedcol'>
              <div>
                <img src={dogedge} className='processed' />
              </div>
            </Col>
          </Row>
          <Row className='result'>
            <div className='resultn'>
              <p className='resultnn'>{this.state.message}</p>
              {this.alertfunc()}

            </div>
          </Row>
          <Row className='backbtn'>
            <div className='backbtn'>
              <Button variant="primary" onClick={this.backfunc}
className='backbtnn'>Back</Button>
            </div>

          </Row>
          <Row className='backbtn'>
            <div className='backbtn'>
              <Button variant="primary" onClick={this.qfunc}
className='backbtnn'>Questionnaire</Button>
            </div>

          </Row>
        </Container>
      </div>
    );
  }
}

export default Disease;
```

- Implementation of Chatbot response

```
import React from 'react';
import PropTypes from 'prop-types';

function AnswerOption(props) {
  return (
    <li className="answerOption">
      <input
        type="radio"
        className="radioCustomButton"
        name="radioGroup"
        checked={props.answerType === props.answer}
        id={props.answerType}
        value={props.answerType}
        disabled={props.answer}
        onChange={props.onAnswerSelected}
      />
      <label className="radioCustomLabel"
htmlFor={props.answerType}>
        {props.answerContent}
      </label>
    </li>
  );
}

AnswerOption.propTypes = {
  answerType: PropTypes.string.isRequired,
  answerContent: PropTypes.string.isRequired,
  answer: PropTypes.string.isRequired,
  onAnswerSelected: PropTypes.func.isRequired
};

export default AnswerOption;
```

```
import React from 'react';
import PropTypes from 'prop-types';

function Question(props) {
  return <h2 className="question">{props.content}</h2>;
}
```

```
Question.propTypes = {
  content: PropTypes.string.isRequired
};

export default Question;
```

```
import React from 'react';
import PropTypes from 'prop-types';

function QuestionCount(props) {
  return (
    <div className="questionCount">
      {/* Question <span>{props.counter}</span> of
<span>{props.total}</span> */}
    </div>
  );
}

QuestionCount.propTypes = {
  counter: PropTypes.number.isRequired,
  total: PropTypes.number.isRequired
};

export default QuestionCount;
```

```
import React from 'react';
import PropTypes from 'prop-types';
import { CSSTransitionGroup } from 'react-transition-group';
import Question from '../components/Question';
import QuestionCount from '../components/QuestionCount';
import AnswerOption from '../components/AnswerOption';
import { Button } from 'react-bootstrap'
import ReactDOM from 'react-dom';
import Quizeapp from '../Quizeapp';
function Home() {


  const refreshPage = ()=>{
    //  window.location.reload();
```

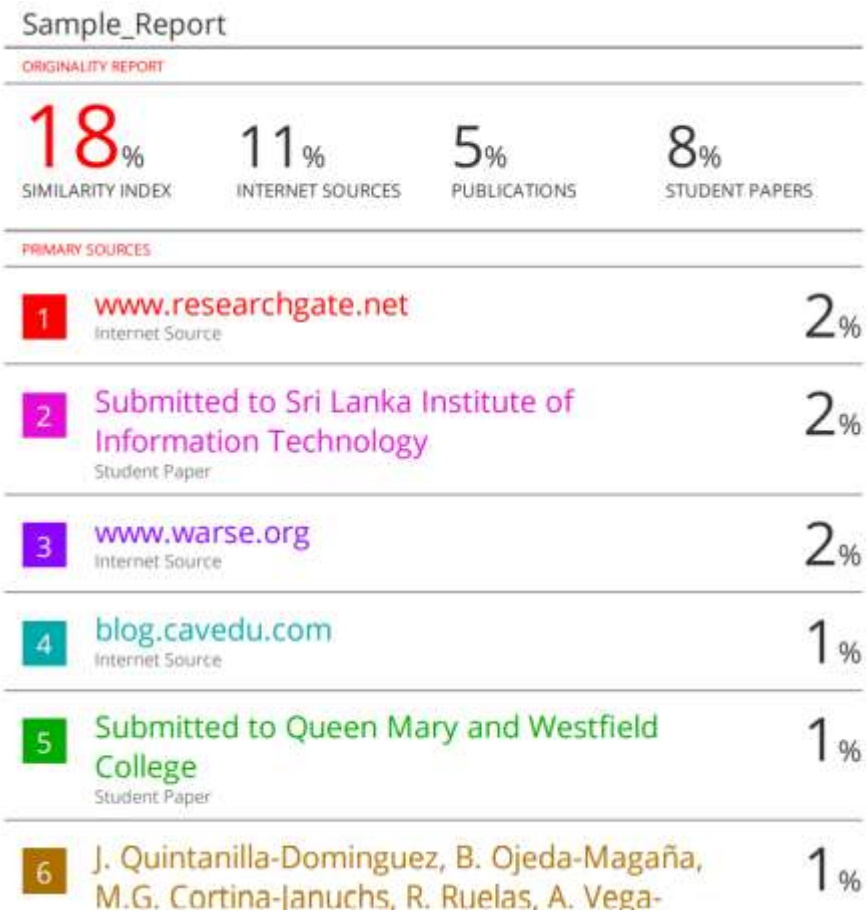
```
     ReactDOM.render(<Quizeapp />,
document.getElementById('root'));
  }
  try{
    return (
      <div id = "buttonmain">
        <button style={{color: 'white', marginLeft: '90%'}}
class="button" onClick={refreshPage}>Home</button>
      </div>
    );
  } catch(e) { console.log('Error')}

}

function Quiz(props) {
  function renderAnswerOptions(key) {
    return (
      <AnswerOption
        key={key.content}
        answerContent={key.content}
        answerType={key.type}
        answer={props.answer}
        questionId={props.questionId}
        onAnswerSelected={props.onAnswerSelected}
      />
    );
  }

  return (
    <CSSTransitionGroup
      className="container"
      component="div"
      transitionName="fade"
      transitionEnterTimeout={800}
      transitionLeaveTimeout={500}
      transitionAppear
      transitionAppearTimeout={500}
    >
      <div key={props.questionId}>
        <QuestionCount counter={props.questionId}
total={props.questionTotal} />
        <Question content={props.question} />
        <ul className="answerOptions">
```

```
          {props.answerOptions.map(renderAnswerOptions)}
        </ul>
      </div>
      {/* <Home/> */}

    </CSSTransitionGroup>
  );
}

Quiz.propTypes = {
  answer: PropTypes.string.isRequired,
  answerOptions: PropTypes.array.isRequired,
  question: PropTypes.string.isRequired,
  questionId: PropTypes.number.isRequired,
  questionTotal: PropTypes.number.isRequired,
  onAnswerSelected: PropTypes.func.isRequired
};

export default Quiz;
```

```
import React from 'react';
import PropTypes from 'prop-types';
import { CSSTransitionGroup } from 'react-transition-group';

function Home() {


  const refreshPage = ()=>{
     window.location.reload();
  }
  try{
    return (
      <div id = "buttonmain">
        <button style={{color: 'white', marginLeft: '90%'}}
class="button" onClick={refreshPage}>Home</button>
      </div>
    );
  } catch(e) { console.log('Error')}

}

function Result(props) {
```

```
  return (
    <CSSTransitionGroup
      className="container result"
      component="div"
      transitionName="fade"
      transitionEnterTimeout={800}
      transitionLeaveTimeout={500}
      transitionAppear
      transitionAppearTimeout={500}
    >
      <div>
      <strong>{props.quizResult}</strong>
      </div>
      {/* <Home/> */}
    </CSSTransitionGroup>
  );
}

Result.propTypes = {
  quizResult: PropTypes.string.isRequired
};

export default Result;
```

Appendices – C



Sample_Report

ORIGINALITY REPORT

18%	11%	5%	8%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.researchgate.net Internet Source	2%
2	Submitted to Sri Lanka Institute of Information Technology Student Paper	2%
3	www.warse.org Internet Source	2%
4	blog.cavedu.com Internet Source	1%
5	Submitted to Queen Mary and Westfield College Student Paper	1%
6	J. Quintanilla-Dominguez, B. Ojeda-Magaña, M.G. Cortina-Januchs, R. Ruelas, A. Vega-	1%