VERSATILE AND REAL-TIME IOT BASED HEALTH MONITORING
SYSTEM AND IDENTIFYING AILMENTS OF THE SKIN

A PROJECT REPORT

Submitted by

DEEPASREE M[211420104055]

DEEPIKA P[211420104056]

INDHUMATHI V[211420104100]

in partial fulfillment for the award of the degree
of
BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

MARCH 2024

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

BONAFIDE CERTIFICATE

Certified that this project report **“VERSATILE AND REAL-TIME IOT BASED HEALTH MONITORING SYSTEM AND IDENTIFYING AILMENTS OF THE SKIN”**work“**DEEPASREE.M[211420104055],DEEPIKA.P[211420104056], INDHUMATHI.V[211420104100]”**whocarried out the project work under my supervision.

Signature of the HOD with date

Dr. L.JABASHEELA M.E., Ph.D.,

PROFESSOR
HEAD OF THE DEPARTMENT

Department of Computer Science and Engineering,
Panimalar Engineering College,

Chennai - 123

Signature of the Supervisor with date

Dr.V.SATHIYA M.E,Ph.D.,

SUPERVISOR
ASSOCIATE PROFESSOR

Department of Computer Science and Engineering,

Panimalar Engineering College,

Chennai - 123

Certified that the above candidate(s) was examined in the End Semester Project

Viva-Voce Examination held on ....

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION BY THE STUDENT

We **DEEPASREE M[211420104055],DEEPIKA P[211420104056],INDHUMATHI V[211420104100]** hereby declare that this project report titled " **VERSATILE AND REAL-TIME IOT BASED HEALTH MONITORING SYSTEM AND IDENTIFYING AILMENTS OF THE SKIN**, under the guidance of **Dr.V.SATHIYA M.E.,Ph.D.,** is the original work done by us and we have not plagiarized or submitted to any other degree in any university by us.

DEEPASREE M[211420104055]
DEEPIKA P[211420104056]
INDHUMATHI V[211420104100]

ACKNOWLEDGEMENT

Our profound gratitude is directed towards our esteemed Secretary and Correspondent, **Dr. P. CHINNADURAI, M.A., Ph.D**., for his fervent encouragement. His inspirational support proved instrumental in galvanizing our efforts, ultimately contributing significantly to the successful completion of this project

We want to express our deep gratitude to our Directors, **Tmt. C. VIJAYARAJESWARI, Dr. C. SAKTHI KUMAR, M.E., Ph.D., and Dr. SARANYASREE SAKTHI KUMAR, B.E., M.B.A., Ph.D.,** for graciously affording us the essential resources and facilities for undertaking of this project.

Our gratitude is also extended to our Principal, **Dr. K. MANI, M.E., Ph.D.,** whose facilitation proved pivotal in the successful completion of this project.

We express our heartfelt thanks to **Dr. L. JABASHEELA, M.E., Ph.D.,** Head of the Department of Computer Science and Engineering, for granting the necessary facilities that contributed to the timely and successful completion of project.

We would like to express our sincere thanks to **Dr.KAVITHA SUBRAMANI,M.E.,Ph.D.,** and **Dr.V.SATHIYA,M.E.,Ph.D.,** and all the faculty members of the Department of CSE for their unwavering support for the successful completion of the project.

DEEPASREE M[211420104055]

DEEPIKA P[211420104056]

INDHUMATHI V[211420104100]

ABSTRACT

The Internet of Things (IoT)-based health monitor is designed to make sick and tired people's mobility and strain easier. It describes a vast global network of physical objects that are connected to the internet .The Internet of Things is at the forefront of technological innovation, reshaping how we interact with surroundings. The objects are implanted with sensors, actuators, and connectivity, which allows them to gather, share, and use data. It has the potential to revolutionize a wide range of industries by providing hitherto unheard- of possibilities for automation, real-time monitoring, and data-driven decision-making. As they can analyze data and employ AI to carry out specified tasks, these gadgets are frequently referred to as "smart devices". Currently, in order to receive outputs, A real-time health monitoring system enables patients to expeditiously finish all required medical testing in one place. Unlike previous research, this study was innovative in that all of the testing devices were used together. The majority of patients would prefer to use these kinds of devices for diagnosis, according to the survey. Multiple devices (blood pressure, heart sensor, metal sensor, alcohol sensor, GSM, GPS) are combined to form a single device. After each circuit is actually put into practice using a variety of sensors and the Arduino platform. Ultimately, discrete circuits were built and assembled on a bread board. Its main objective is also to obtain all of the test data from the server. All of the test results were transferred from the main device to the web server and skin illness prediction was carried out, and the result is viewed simultaneously.The accuracy for the health monitoring system and skin disease prediction is 98%. An application is built which is used to predict skin disorders using machine learning, analyze the skin diseases, and report on them. After the primary device was put into use and measured, the results obtained were deemed to be fairly satisfactory. After the system's efficacy was investigated, several cost- effective and human health-safe results were reached. In light of the results of the final survey, which aims to collect feedback from device users, some conclusions might be suggested.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

1.1 OVERVIEW

IOT refers to a broad worldwide network of physical things with internet connections that may collect and exchange data. devices are sometimes called "smart devices" because they have the ability to analyse data and use artificial intelligence (AI) to perform certain tasks. IOT devices continuously record and store user and performance- related data. As such, it can be used to skin problems, challenges, and trends as well as make recommendations. Due to this state of affairs, several, often contradictory suggestions have been made for the proper operation of IOT systems. Because of its originality and intricacy, the centre of interest of an IOT network, along with the necessary backend nearby selections and gadgets, lacks a mounted fantastic workout from a computer standpoint. The Internet of Things is made up of sensors, processors, and microcontrollers along with other components that are used for communication over the internet. It is designed with an appropriate protocol that facilitates user and peer interaction .The Internet of Things has a big impact on healthcare, which makes things easier for doctors and patients. The effective healthcare system offers preventive in place of pricey clinical care, and homecare is offered in its substitute. The service will be beneficial, since it adheres to basic healthcare, which produces more favorable outcomes. The cost and quality of patient care are being supported by IOT technology.

Sensors are an IoT device's fundamental parts. Device serve as a layer of interface between the Internet of Things system and the outside world. The essential component that senses the target and collects data is a sensor. Sensor data is the base for all the operations and outcomes of an Internet of Things enabled device. While some sensors are made with a specific objective in mind, others are more universal and can be utilized in a variety of applications. The medical and health care industries are the most alluring and developing fields for IOTs. With the introduction of iot, a wide range of possible medical applications are raised. Providing diagnosis and treatment from home appears to be a growing and likely use. Consequently, a broad range of medical gadgets referred to as "smart medical devices". The health sensor device's default ability to communicate with the server can be used to communicate with a computer or smartphone, which lowers the overall cost and complexity of the system. IoT demonstrates a trustworthy ongoing medical monitoring system that allows patients' solutions, no matter where in the globe, to be verified. In order to gather data about their bodies.

1.2 PROBLEM DEFINITION

The use of sensor-equipped devices for dermatological evaluations may be made easier by the growth of telemedicine platforms and remote patient monitoring programs. It would allow people to obtain prompt advice and comments from medical specialists without having to make in-person visits. Sensors are essential parts of many gadgets, from wearables to smartphones and more. Though sensors have been widely used in wellness tracking and health monitoring, especially in wearables, they usually lack specific features for predicting skin diseases. The prognosis of skin diseases frequently depends on clinical examination, specialized diagnostic instruments, and occasionally imaging methods like dermatoscopy. Sensor technology has proliferated, but it is still largely the realm of medical specialists and specialized equipment to diagnose and anticipate skin illnesses, rather than being incorporated into consumer-grade products. Future approaches to skin care and disease prevention might be completely changed by sensor-enabled gadgets, which can bridge the gap between consumer accessibility andmedical competence.

1.3 OBJECTIVE

Convenience and accessibility for users will be prioritized in the design of the Arduino-based health monitoring system. Because of its small size and portability, users will be able to easily incorporate it Wireless connectivity options will be included in the system, allowing data to be transmitted through sensors for easy tracking and analysis. Using image processing techniques, the skin ailment identification module will be able to precisely diagnose a variety of dermatological illnesses, ranging from common problems like eczema and acne to more serious ones like melanoma. Early detection capabilities enable users to take preventative action and swiftly address skin concerns, potentially averting consequences and enhancing overall skin health.The system will include gamification features and tailored feedback systems to improve user engagement and adherence to health monitoring habits. Users will be motivated to continue being proactive about their health and to maintain regular monitoring practices through interactive features and encouraging cues. Overall, by offering thorough insights into both their physiological indicators and skin state, the integration of these technologies will enable customers to take control of their health and well-being. The goal of the Arduino-based health monitoring system is to promote proactive preventive treatment and revolutionize person

CHAPTER 2

LITERATURE REVIEW

Advancements in healthcare technology have spurred the development of integrated systems that combine health monitoring and skin disease detection. A comprehensive review of pertinent literature reveals key insights into the integration of various technologies, including the Internet of Things (IoT), machine learning, and image processing, aimed at enhancing healthcare diagnostics and monitoring.The foundational work in health monitoring systems is evident in the study titled "Health Monitoring Based on Internet of Medical Things" [4]. This research elucidates the architecture and enabling technologies crucial for real-time health monitoring. The study emphasizes the significance of incorporating IoT principles into health monitoring systems and underscores the potential for continuous
and remote health tracking

Simultaneously, "IOT BASED HEALTH MONITORING SYSTEM" [8] introduces an IoT-centric health monitoring paradigm, delving into the integration of IoT for dynamic and personalized health assessment. This research contributes to the establishment of a framework for leveraging cutting-edge technologies in health monitoring. The collective findings of these studies lay the groundwork for the implementation of advanced approaches in real-time health monitoring, with a focus on the integration of IoT principles to enhance the dynamic and personalized assessment of an individual's health.In the domain of dermatological diagnostics, the integration of deep learning methodologies has become increasingly pronounced, as exemplified in the study titled "Automatic Diagnosis of Skin Diseases Using Convolutional Neural Network" [13]. The focal point of this research lies in the utilization of convolutional neural networks (CNNs) to streamline the diagnostic process for various skin conditions. By leveraging the capabilities of artificial intelligence, the study underscores the considerable potential for
automating and enhancing dermatological assessments.

Similarly, another significant contribution to the field is found in the research titled "A machine learning model for skin disease classification using convolution neural network" [14]. This study propels the application of machine learning algorithms in the classification of skin diseases, further substantiating the pivotal role of computational intelligence in the realm of dermatology. The emphasis here is on advancing the accuracy and efficiency of skin disease classification through the implementation of convolutional neural networks, highlighting the transformative impact of machine learning on traditional diagnostic approaches.the integration of advanced technologies is prominently featured in the research paper "Skin Disease Classification from Image" [12]. This study employs sophisticated image processing techniques to address challenges in image-based identification of skin diseases, contributing to the knowledge base by presenting an innovative method for classifying skin diseases from images. The research highlights the potential of image processing in enhancing diagnostic accuracy.

In parallel, "Advance study of skin diseases detection using image processing methods" [3] delves into the exploration of advanced image processing methods for skin disease detection. This study provides nuanced insights into the intricacies of image-based identification techniques, contributing valuable knowledge for the refinement and optimization of skin disease detection processes. The research focuses on the comprehensive examination of sophisticated image processing techniques, emphasizing the significance of continuous exploration and enhancement in the ongoing quest to improve diagnostic capabilities in the field of dermatology.

In synthesis, these studies collectively underscore the transformative potential of integrated health monitoring systems and skin disease detection. The convergence of IoT, machine learning, and image processing technologies promises a paradigm shift in healthcare, offering more personalized, efficient, and accurate diagnostic and monitoring solutions. As these technologies continue to evolve, the envisioned integrated systems hold promise for revolutionizing

CHAPTER 3
THEORETICAL BACKGROUND

3.1 EXISTING SYSTEM

Health monitoring utilizing Arduino and skin disease prediction exhibit notable advantages, but they are not without their challenges. In health monitoring systems based on Arduino, limitations in processing capability may hinder the implementation of advanced algorithms or the management of extensive health datasets. Additionally, concerns about accuracy in vital sign measurements may arise due to sensor quality variations. The system are designed separately for heart beat, Temperature, Blood pressure which may leads to overpricing and also time consuming. Scalability challenges could restrict the adaptability of Arduino-based solutions to evolving healthcare needs. The collected data is then sent to a nearby database server which is used to organize, index, and transport data when needed all personnel are able to access records via an interface software program. On the other hand, the application in skin disease prediction systems introduces limitations in computational power, potentially impacting the processing of complex artificial intelligence models crucial for accurate predictions. Image resolution constraints may compromise the system's ability to discern subtle details necessary for precise skin disease identification. The system is limited due to a lack of location context, the resulting system would therefore not be able to indicate patient location in the event user vitals suggest aid was required

3.2 PROPOSED METHODOLOGY

The state-of-the-art Health Monitoring System seamlessly incorporates Arduino-based sensors to collect physiological data continuously and record important metrics such as activity levels, temperature, and heart rate. Concurrently, data on skin conditions is analysed by Flask-powered Skin Disease Prediction module using machine learning techniques. The

Arduino component of the system guarantees precise, real- time health monitoring, making it simple for people to track their well-being. Based on Flask's versatility, the Skin Disease Prediction module provides individualized insights into probable dermatological disorders on an easy-to-use homepage. the suggested solution promises to transform personal health management by offering thorough skin health evaluations and timely alarms using a safe and scalable architecture. This all- encompassing strategy encourages a new era of easily accessible, data-driven healthcare by enabling consumers to proactively address any health issues.

3.3 FEASIBILTY STUDY

A feasibility study is an evaluation and analysis of a project's potential based on thorough investigation and research to provide decision-makers complete confidence. The goal of feasibility studies is to logically and objectively identify the advantages and disadvantages of a potential project or existing business, as well as the possibilities and dangers presented by the surrounding environment, the resources required to execute the plan, and, ultimately, the likelihood of success. The most important things to think about are: There are three levels of feasibility:1. economic,2. technical

3.3.1 ECONOMIC FEASIBILITY

Economic feasibility is a crucial aspect of a feasibility study that assesses the financial viability and potential economic benefits of a proposed project. It involves analysing the costs associated with the project, as well as estimating the potential returns on investment. The primary goal is to determine whether the project is financially feasible and economically justifiable.

Potential Revenue Streams:

Purpose: To explore avenues for offsetting development and operational costs and achieving financial sustainability.

Return on Investment (ROI) Analysis:

Purpose: To assess the financial attractiveness of the project and its potential impact on healthcare outcomes.

Cost-Benefit Analysis (CBA):

Purpose: To provide decision-makers with a clear understanding of the economic value and feasibility of the project.

Payback Period Analysis:

Purpose: To assess the project's risk and provide insights into its financial payback period.

3.3.2 TECHNICAL FEASIBILITY

T he proposed system demonstrates robust technical feasibility by seamlessly integrating hardware andsoftware components. The Arduino-based health monitoring sensors ensure real-time data accuracy, reliability, and low-cost implementation. The Flask framework, chosen for its versatility, powers Skin Disease Prediction module, allowing efficient deployment and customization. The system leverages widely-used technologies, ensuring compatibility with various devices and platforms. Its modular design facilitates easy scalability and integration with emerging technologies. The technical infrastructure supports secure data transmission, storage, and retrieval, adhering to healthcare data standards. With a focus on user-friendly interfaces and responsive web design, the system guarantees an intuitive experience for both health monitoring and skin disease prediction. The technical feasibility of the proposed system positions it as an adaptable, future- proof solution for personalized health management.

1. Artificial Intelligence
2. programming language: python ,html, css, embedded c
3. INO file:health monitoring system(Arduino)
4. Integrated Development Environment :Vscode,Arduino IDE

3.4 IMPLEMENTATION ENVIRONMENT

Hardware Setup:

Arduino microcontroller board (Arduino Uno)

Sensors for vital signs (temperature, blood pressure, metal detector, gas detector)

Software Setup:

Programming

Language:python,css,html,embeded c

Technology: Artificial Intelligence

Operating

System:windows 10

Software:VScode

.INO file: health monitoring system

Framework: flask,keras,tensorflow

CHAPTER 4
SYSTEM DESIGN

4.1 FLOW DIAGRAM

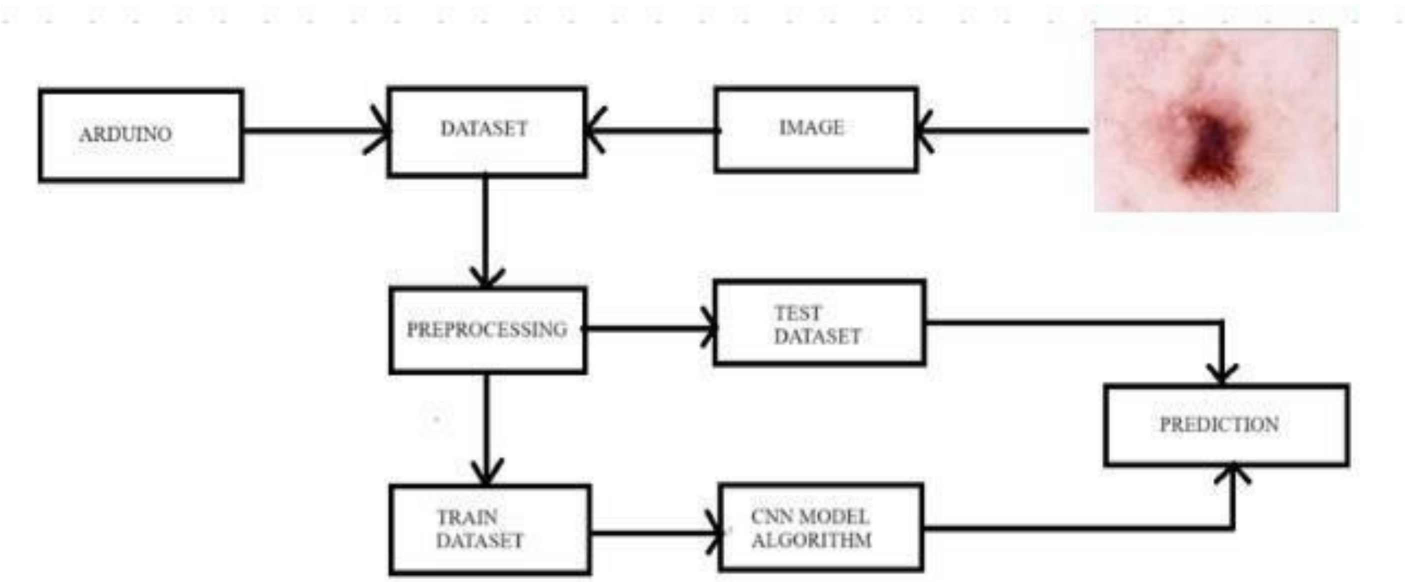

FIG 4.1.1 FLOWDIAGRAM

The skin disease prediction and health monitoring system aim to provide users with an efficient tool for early detection of skin conditions and continuous health monitoring. The flow diagram outlines the sequential steps involved in the system, emphasizing its user-friendly interface and comprehensivefunctionalities.

4.2 DATASET DESCRIPTION

The Skin Disease Image Dataset is a curated collection of high-resolution images aimed at facilitating research and development in the field of dermatology and healthcare. The purpose of this dataset is to aid in the development and assessment of machine learning models that forecast different skin diseases from visual data.

FIG 4.2.1 DATASET DESCRIPTION

S.NO	CLASS	NO OF IMAGES
01	BA- cellulitis	136
02	BA-impetigo	156
03	FU-athlete-foot	122
04	FU-nail-fungus	172
05	FU-ringworm	135
06	PA-cutaneous-larva-migrans	144
07	VI-shingles	123
08	Actinic Keratoses	153
09	Vascular lesion	146
10	Benign Keratosis	177
11	Dermatofibroma	198
12	Melanoma	155

CHAPTER 5

SYSTEM ARCHITECTURE

5.1 ARCHITECTURE OVERVIEW

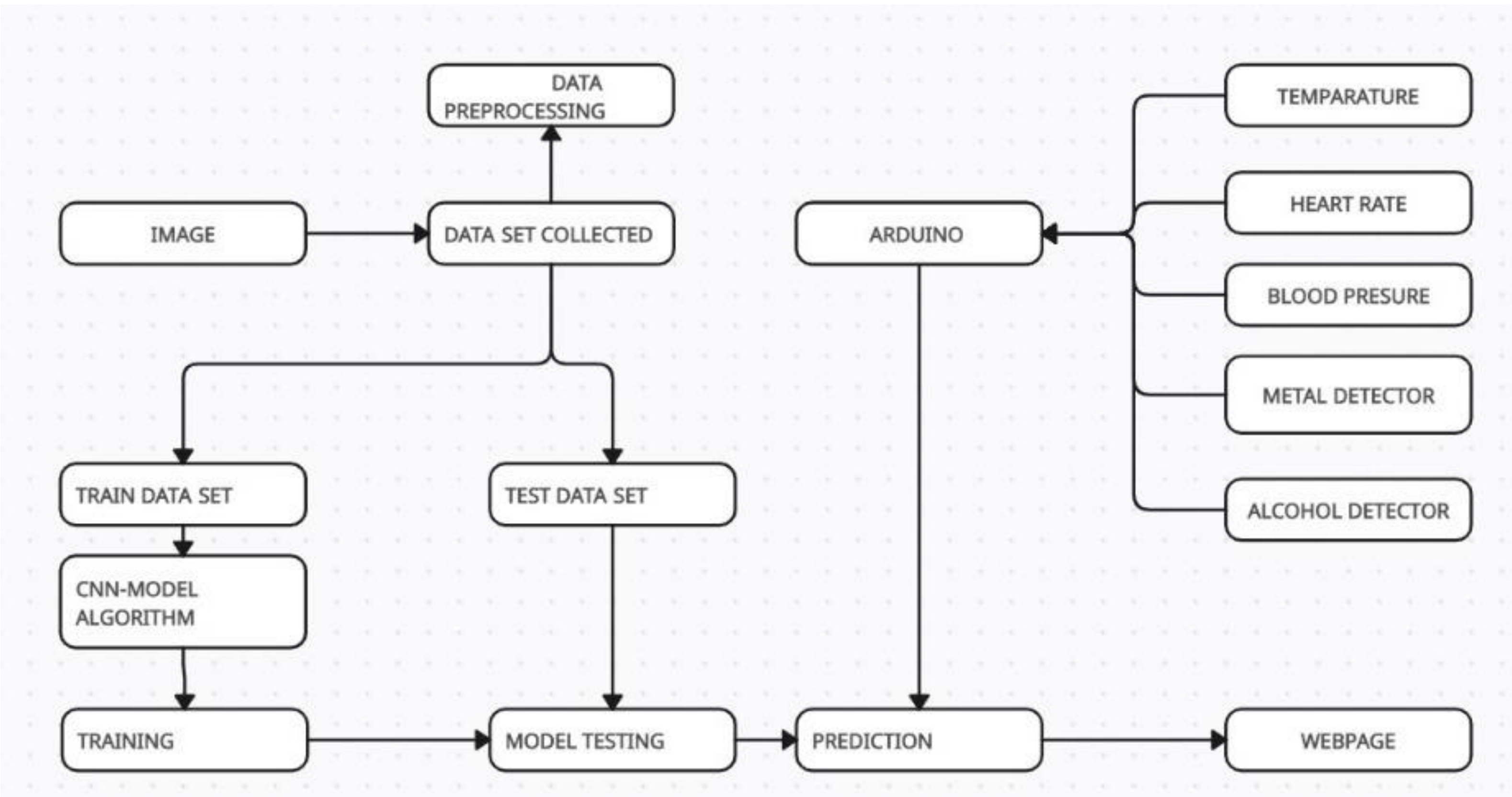

FIG 5.1.1 ARCHITECTURE DIAGRAM

The skin disease prediction and health monitoring system aim to provide users with an efficient tool forearly detection of skin conditions and continuous health monitoring.

IMAGE

The dataset comprises a diverse set of high-quality images capturing different skin conditions. Each image is labeled with the corresponding skin disease for supervised learning tasks. To facilitate thorough analysis and feature extraction for machine learning, all images have high-quality pixels.

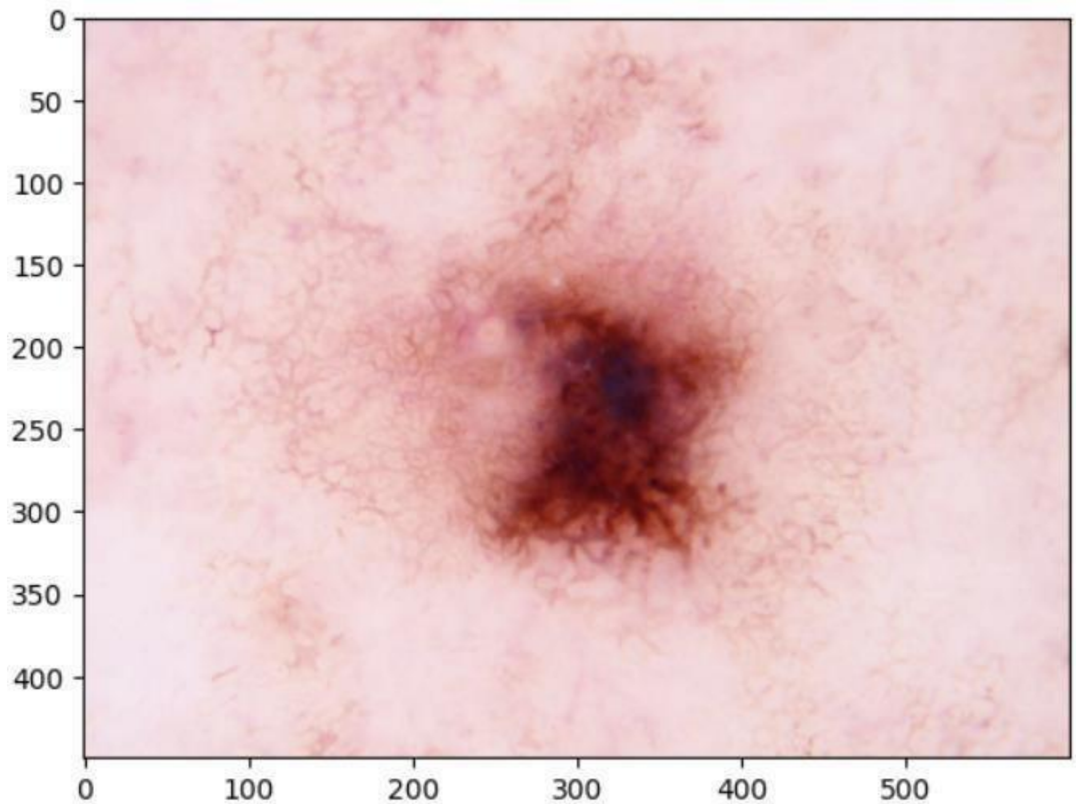

FIG 5.1.2 IMAGE DATASET

PREPROCESSING

Machine learning models to predict skin diseases from picture datasets, preprocessing is an essential step in the process. This initial stage consists of a number of procedures intended to improve the relevance and quality of the data, making it suitable for training reliable and accurate models. picture normalisation, scaling, and feature extraction are common preparation procedures used in the context of skin disease picture datasets. By standardising pixel values, normalisation ensures uniformity across images and guards against biases in the training of models. In order to preserve consistency in the dimensions of images and enable effectivecomputing during model training, resizing is essential.

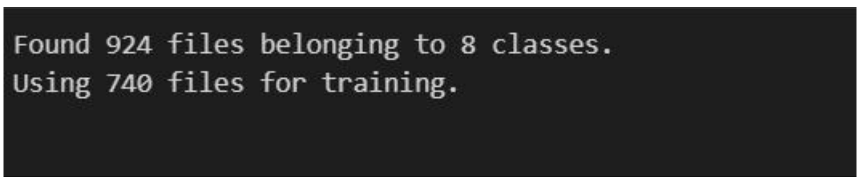

FIG 5.1.3 PREPROCESSING

TRAINING AND TESTING

Skin Disease Prediction and Health Monitoring System using Convolutional Neural Networks (CNNs) involves a systematic process beginning with data collection and preprocessing. A diverse and labeled dataset of skin images is acquired, followed by the division of data into training and testing sets. The CNN architecture is then designed, considering factors such as layer configuration and activation functions. Data augmentation techniques are applied to enhance model generalization, and the model is compiled using appropriate loss functions and optimizers. Subsequently, training is conducted, monitoring key metrics like accuracy and loss. The trained model is evaluated on a testing dataset, and fine-tuning may be performed for optimization.

FLASK

Flask is a lightweight and flexible web framework for Python that is commonly used for building web applications.The Flask application for health monitoring and skin disease prediction aims to provide users with a seamless and user-friendly platform to track their health metrics and receive predictions related to skin

conditions. Leveraging Flask's capabilities, the application offers a range of features including user authentication, data visualization, machine learning integration.Flask's routing system enables the creation of endpoints to handle different HTTP requests. Views are implemented using Python functions, allowing developers to define the logic for processing requests and generating responses.Flask-Login extension facilitates user authentication and session management. It enables features such as user registration, login, logout, and password reset, ensuring secure access to the application.Flask integrates with libraries such as Plotly or Matplotlib to visualize health metrics data in the form of charts and graphs. These visualizations aid users in understanding their health trends and patterns effectively.Flask serves as the backend for integrating machine learning models trained on skin disease datasets. Users can submit images or descriptions of skin conditions, and Flask handles the prediction process, providing relevant results to the user.Flask provides mechanisms for error handling and logging, ensuring robustness and monitoring of the application. Custom error handlers can be implemented to gracefully handle exceptions and provide meaningful error messages to users.The Flask application for health monitoring and skin disease prediction demonstrates the effectiveness of Flask in building scalable and feature-rich applications catering to the healthcare domain.Flask is suitable for the backend of a skin disease prediction application, the frontend (HTML, CSS, and JavaScript) and the machine learning model itself are crucial components that should be developed and integrated accordingly. Additionally, ensure that the application complies with relevant data privacy and security standards, especially when dealing with sensitive medical information.

- Routing

- Templates
- Request Handling
- Response Handling
- Static Files

ARDUINO

The Arduino board, has a microcontroller that carries out preprogrammed instructions. Arduino boards come in a variety of variants, each designed to meet particular requirements. They have microcontrollers, USB ports, and a variety of input/output (I/O) pins. Together, these parts give users the ability to connect actuators, sensors, and other electronic devices, which makes the Arduino a flexible platform for creative expression.the Arduino- based health monitoring system by inputting data via body sensors.

SENSORS

The sensors that measure blood pressure, temperature, heart rate, metal detector, and alcohol detector arelinked to the Arduino board.

TEMPERATURE SENSOR

The sensor is used to manipulate and calculate the temperature with an output voltage that is valued with theCentigrade temperature. The sensor used is LM35 sensor which has a clinical advantage over the regular based linear temperature sensor, as the client doesn't convert the scale between Kelvin and Centigrade due tothe fact LM-35 calibrates the difference right away and has higher efficiency than thermistor. Normal bodytemperature of human body is 98 *F.HHM has a unique feature that it will decide whether temperature is normal or abnormal if temperature is above 100 it will print abnormal below 100 it will print normal.

FIG 5.1.4 TEMPERATURE SENSOR

HEART BEAT SENSOR

It is as a degree for the coronary heart beat of the affected man or woman. It offers a digital output of coronary heart based beat when the patient uses their finger which is positioned on the sensor. This compressed device sensor uses an operating voltage of +5V DC. The calculation is based on the flow of bold over the Heart beat sensor, used to measure the beat which usually lies among 60-100 bpm

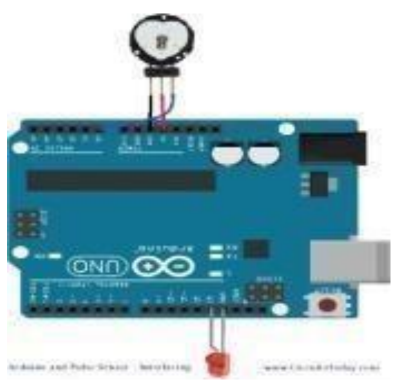

FIG 5.1.5 HEART BEAT SENSOR

BLOOD PRESSURE SENSOR

The sensor is designed and used to measure the stress of human blood. Sensor that is used to measure the stress based on systolic and diastolic, which has higher accuracy and reliable than the traditional sphygmomanometer, A tool which is based on air bladder cuff used with a basic stethoscope to measure the blood pressure in an artery of the human body**.**

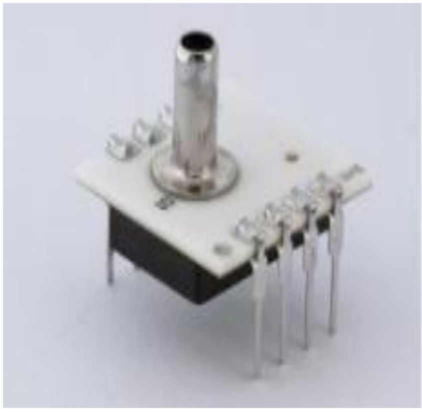

FIG 5.1.6 BLOOD PRESSURE SENSOR

ALCOHOL SENSOR

Gas sensors are to be had in huge specifications depending on the sensitivity tiers, sort of gas to be sensed, physical dimensions with other features. The device is based on a cover with a methane based fuel sensor used to detect the gasses along with ammonia leading to the formation of methane eventually. Upon interaction with the sensor, ionization takes place between the materials leading to absorption by the sensing metal, causing an observable potential distinction sent to the processor unit via the output pins.

FIG 5.1.7 GAS SENSOR

METAL DETECTOR

A metal detector is a virtual device that is used to detect the metallic presence based on what is found within the vicinity. A detector is used for location of multiple metallic components hidden within internal objects, or used as a metallic gadget buried deep underground the crust of the earth. A collection of items are collected when swept over the ground as multiple objects are buried beneath the surface.

FIG 5.1.8 METAL SENSOR

5.2 CNN ALGORITHM

A Convolutional Neural Network (CNN) algorithm operates by leveraging convolutional layers, which apply a set of learnable filters to small regions of the input image, capturing different features like edges, textures, or patterns. These convolutions are followed by non-linear activation functions such as ReLU to introduce non-linearity and enhance the model's capacity to learn complex representations. Subsequently, pooling layers are employed to reduce the spatial dimensions of the feature maps while preserving relevant information. This hierarchical feature extraction process enables the network to learn progressively abstract features. The learned features are then flattened and passed through fully connected layers for higher-level reasoning and decision-making, ultimately leading to the classification or detection of objects in the input image. During training, the network's parameters are iteratively adjusted using optimization algorithms like gradient descent, minimizing a predefined loss function, such as cross-entropy loss, to improve the network's performance in accurately predicting the target labels associated with the input images. This iterative learning process allows CNNs to automatically learn discriminative features from raw input data, making them highly effective in various visualrecognition tasks across different domains.

IMAGE RESIZING:

Resized Image=resize(Original Image,(Target Width,Target Height))

NORMALIZATION:

Normalized Pixel Value=Original Pixel Value / Maximum Pixel Value

Model: "sequential_5"

Layer (type)	Output Shape	Param #
conv2d_12 (Conv2D)	(None, 238, 238, 8)	224
conv2d_13 (Conv2D)	(None, 236, 236, 16)	1168
max_pooling2d_9 (MaxPooling2D)	(None, 118, 118, 16)	0
dropout_10 (Dropout)	(None, 118, 118, 16)	0
conv2d_14 (Conv2D)	(None, 116, 116, 32)	4640
max_pooling2d_10 (MaxPooling2D)	(None, 58, 58, 32)	0
conv2d_15 (Conv2D)	(None, 56, 56, 32)	9248
max_pooling2d_11 (MaxPooling2D)	(None, 28, 28, 32)	0
dropout_11 (Dropout)	(None, 28, 28, 32)	0
flatten_3 (Flatten)	(None, 25088)	0
dense_7 (Dense)	(None, 64)	1605696
dropout_12 (Dropout)	(None, 64)	0
dense_8 (Dense)	(None, 9)	585

Total params: 1621561 (6.19 MB)

FIG 5.2.1 CNN Summary

5.3 MODULE DESCRIPTION

SKIN DISEASE AND HEALTH MONITORING SYSTEM

The Health Monitoring and Skin Disease Prediction module is a critical component of the web development project aimed at providing users with comprehensive health tracking capabilities andpredictive analysis for skin conditions.The skin disease prediction module integrates several key components to facilitate accurate diagnosis and prediction of skin conditions. Leveraging machine learning algorithms trained on diverse datasets, the module processes input data such as images or symptom descriptions provided by users. Utilizing Flask as the web framework, the module employs user authentication for secure access, data visualization to present predictions comprehensively.The skin disease prediction module offers users a reliable platform for early detection and management of skin conditions, thereby promoting proactive healthcare practices. It integrates various functionalities, including real-time health monitoring, data analysis, and machine learning-based prediction

Real-time Health Monitoring

Continuous tracking of vital health metrics such as heart rate, blood pressure, temperature, and otherrelevant parameters.

Skin Disease Prediction

Integration of machine learning algorithms to analyze user-provided skin images or symptomdescriptions and predict potential skin conditions with associated confidence scores.

Data Visualization

Visual representation of health metrics and skin disease prediction results through interactive charts,graphs, and reports for easy interpretation.

It collects and processes health data from various sources, including wearable devices and manual input by users.Utilizes machine learning models trained on skin disease datasets to analyze and predict skin conditions based on user-provided data.Implements algorithms for real-time monitoring of health metrics and detection of anomalies or patterns indicating potential health issues.Integrates with the web application's user authentication and database management systems to ensure secure access and storage of user data.

Dependencies

Python libraries such as Flask, scikit-learn, TensorFlow for web development and machine learningintegration.JavaScript libraries like Plotly.js or D3.js for interactive data visualization.

5.3.1 PRE- REQUISITES

Python packages for skin disease detection. The libraries on the list are:

- Python
- Flask
- Pillow==5.4.1
- gevent==1.4.0
- gunicorn==19.9.0
- keras==2.10.0
- tensorflow==2.10.0
- Numpy
- Softwareserial
- Ethernet
- Arduino

5.3.2 PROJECT FILE STRUCTURE

- Skin disease_dataset - Dataset folder contains the images.
- Model.h5 -the Hierarchical Data Formats (HDF) for storing a lot of data is an H5.Large volumes of data are stored there in the form of multidimensional arrays. The format is mostly used to store organised scientific data that can be quickly retrieved and analysed.
- Model.json -JSON files are used to serialise structured data and send it across a network,

usually from a web application to a server.

- HTML file -A skin disease prediction system's visual interface, which enables users to interact with the system, enter pertinent data, get forecasts, and traverse the system,is created by HTML files.
- These files handle user input, execute predictions, and return results to the user interface in concert with backend components
  The HTML file are used for
  - User Interface (UI)
  - Form Creation
  - File Upload,
  - Result Display.

- CSS file-The skin disease prediction system's visual design and layout are mostly determined by CSS files. They handle accessibility and responsiveness issues and combine with HTML files to produce an aesthetically beautiful and easy-to-use interface.The css file are used for
  - Styling HTML Elements
  - Layout and Positioning
  - Responsive Design
  - Color Scheme
  - Images and Icons
  - Form Styling:

- Js file-JavaScript files enhance the skin disease prediction system by adding interactivity, improving the user experience, and facilitating communication between the client and server. The combination of HTML, CSS, and JavaScript creates a dynamic and responsive web application for predicting and managing skin diseases.The js file are used for
  - Form Validation
  - Image Handling

- App.py-In a Flask-based skin disease prediction system, the app.py file functions as the central script and entry point for the web application. This Python file leverages the Flask framework to create and define the behavior of the application. the app.py file includes the instantiation of a Flask application, the definition of routes, and the integration of a skin disease prediction function.
- .ino file-The '.ino ' file, serving as the primary script in Arduino development, encapsulates the code for a specific project within the Arduino Integrated Development Environment (IDE). This file extension is an identifier for Arduino sketches.
- Train_set and test_set dataset- the resized images will be specified.
- Skindi.ipynb- the training and testing are specified .

CHAPTER 6
PERFORMANCE ANALYSIS

6.1 SYSTEM TRAINING

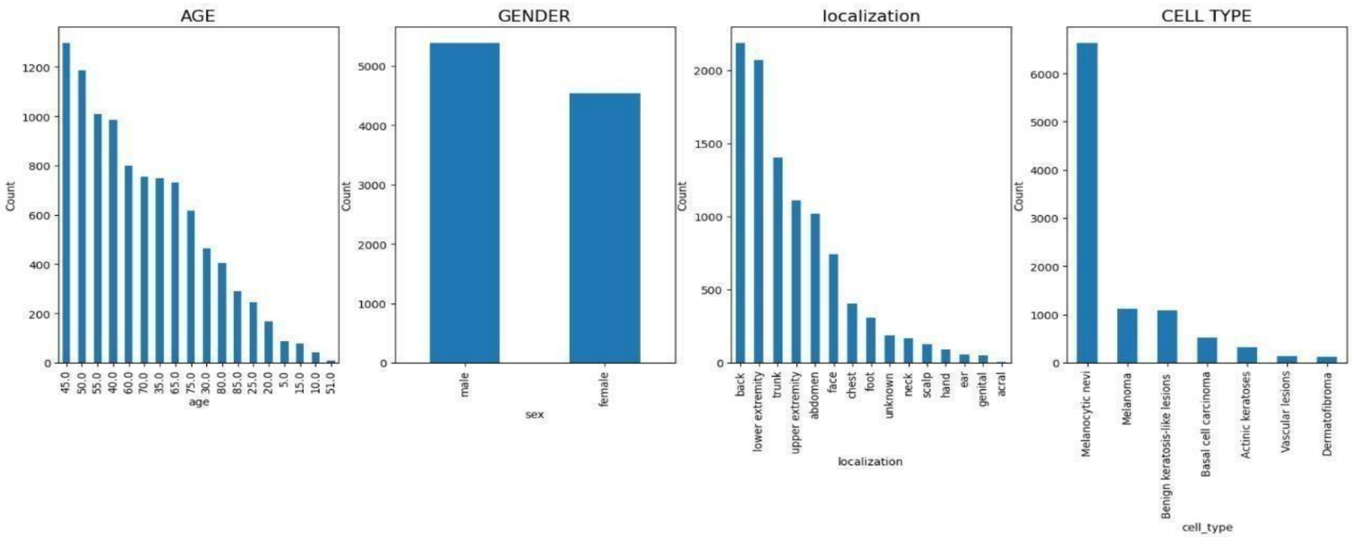

FIG 6.1.1 TRAINING OF EACH ATTRIBUTE

Training a model for skin disease prediction involves training on each attribute, where each attribute might correspond to specific features or characteristics associated with skin

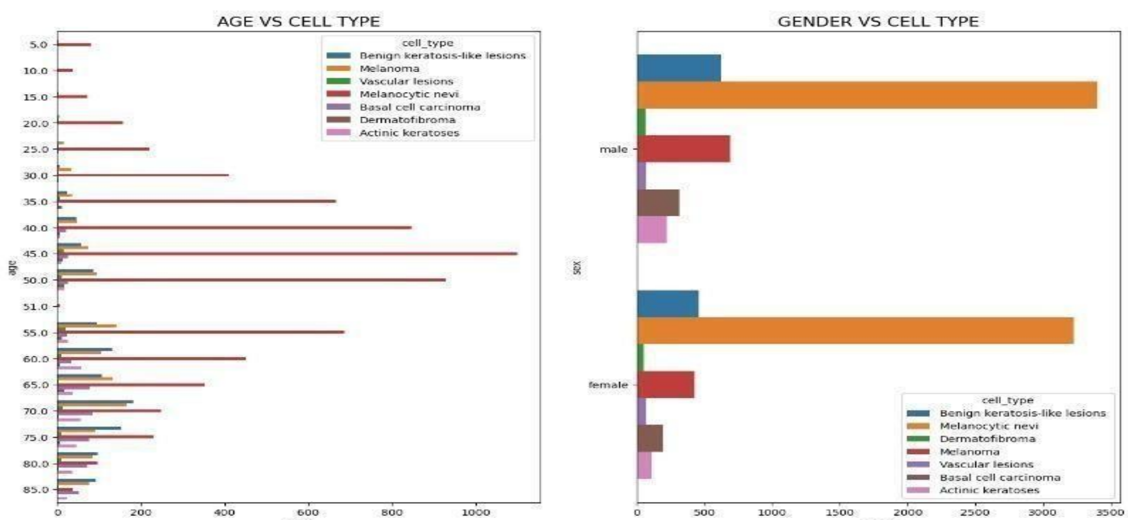

FIG 6.1.2 GRAPH ON COMPARISON

FIG 6.1.3 DATA AUGUMENTATION

Data augmentation enhances CNN models' generalization by increasing training data diversity, preventing overfitting, and recognizing real-world image variations and patterns

FIG 6.1.4 TRAINING ON DATASETS

The process of training a skin disease prediction model begins with the meticulous collection of a diverse and well-annotated dataset comprising images that encompass a spectrum of skin conditions. This dataset serves as the foundation for teaching the model to recognize and classify different dermatological issues accurately.

```
16705208/16705208 [==============================] - 3s 0us/step
Epoch 1/30
24/24 [==============================] - 28s 701ms/step - loss: 0.9910 - accuracy: 0.6527 - val_loss: 0.4248 - val_accuracy: 0.8207
Epoch 2/30
24/24 [==============================] - 14s 592ms/step - loss: 0.3358 - accuracy: 0.8703 - val_loss: 0.3599 - val_accuracy: 0.8696
Epoch 3/30
24/24 [==============================] - 14s 594ms/step - loss: 0.2518 - accuracy: 0.9189 - val_loss: 0.3337 - val_accuracy: 0.8859
Epoch 4/30
24/24 [==============================] - 14s 593ms/step - loss: 0.1839 - accuracy: 0.9446 - val_loss: 0.2848 - val_accuracy: 0.9130
Epoch 5/30
24/24 [==============================] - 14s 598ms/step - loss: 0.1267 - accuracy: 0.9676 - val_loss: 0.2268 - val_accuracy: 0.9457
Epoch 6/30
24/24 [==============================] - 15s 621ms/step - loss: 0.1354 - accuracy: 0.9622 - val_loss: 0.2347 - val_accuracy: 0.9348
Epoch 7/30
24/24 [==============================] - 14s 597ms/step - loss: 0.1225 - accuracy: 0.9541 - val_loss: 0.1832 - val_accuracy: 0.9565
Epoch 8/30
24/24 [==============================] - 14s 593ms/step - loss: 0.1072 - accuracy: 0.9676 - val_loss: 0.1911 - val_accuracy: 0.9511
Epoch 9/30
24/24 [==============================] - 14s 586ms/step - loss: 0.1145 - accuracy: 0.9595 - val_loss: 0.1851 - val_accuracy: 0.9565
Epoch 10/30
24/24 [==============================] - 14s 581ms/step - loss: 0.0816 - accuracy: 0.9770 - val_loss: 0.1980 - val_accuracy: 0.9511
Epoch 11/30
24/24 [==============================] - 14s 584ms/step - loss: 0.0904 - accuracy: 0.9716 - val_loss: 0.2190 - val_accuracy: 0.9402
Epoch 12/30
24/24 [==============================] - 19s 816ms/step - loss: 0.0717 - accuracy: 0.9811 - val_loss: 0.2100 - val_accuracy: 0.9457
Epoch 13/30
24/24 [==============================] - 16s 658ms/step - loss: 0.0839 - accuracy: 0.9716 - val_loss: 0.1637 - val_accuracy: 0.9511
Epoch 14/30
24/24 [==============================] - 15s 640ms/step - loss: 0.0986 - accuracy: 0.9716 - val_loss: 0.2140 - val_accuracy: 0.9402
Epoch 15/30
24/24 [==============================] - 15s 631ms/step - loss: 0.0496 - accuracy: 0.9892 - val_loss: 0.1716 - val_accuracy: 0.9402
Epoch 16/30
24/24 [==============================] - 15s 623ms/step - loss: 0.0535 - accuracy: 0.9878 - val_loss: 0.1442 - val_accuracy: 0.9565
Epoch 17/30
24/24 [==============================] - 15s 630ms/step - loss: 0.0496 - accuracy: 0.9865 - val_loss: 0.1639 - val_accuracy: 0.9565
```

FIG 6.1.5 TRAINING ACCURACY

Training a machine learning model for accuracy involves optimizing the model to correctly predict the target outcomes on the given dataset. Accuracy is a commonly used metric that represents the proportion of correctly classified instances out of the total instances in the dataset.

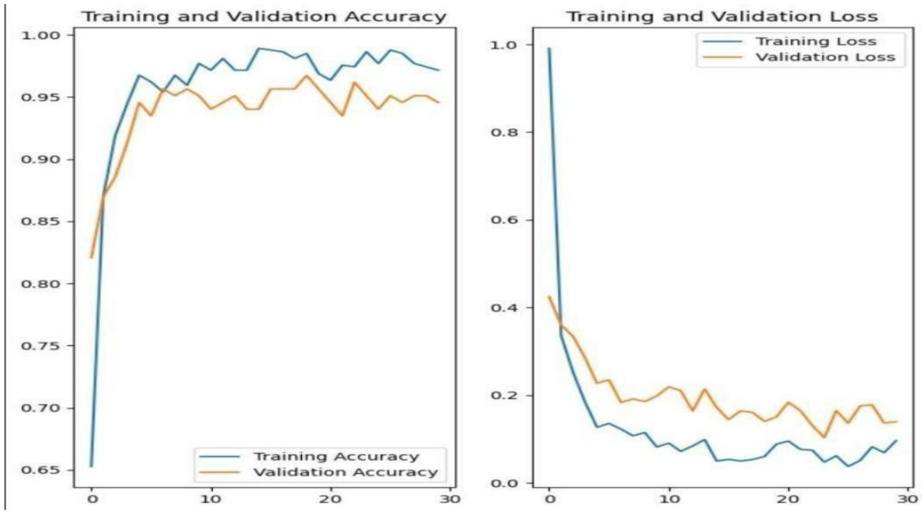

FIG 6.1.6 LINE GRAPH ON TRAINING ACCURACY

VALIDATION

```
Epoch 1/30
24/24 [==============================] - 50s 2s/step - loss: 0.0404 - accuracy: 0.9851 - val_loss: 0.1753 - val_accuracy: 0.9674
Epoch 2/30
24/24 [==============================] - 26s 1s/step - loss: 0.0623 - accuracy: 0.9824 - val_loss: 0.1624 - val_accuracy: 0.9457
Epoch 3/30
24/24 [==============================] - 15s 647ms/step - loss: 0.0510 - accuracy: 0.9811 - val_loss: 0.1445 - val_accuracy: 0.9565
Epoch 4/30
24/24 [==============================] - 15s 618ms/step - loss: 0.0500 - accuracy: 0.9865 - val_loss: 0.1541 - val_accuracy: 0.9620
Epoch 5/30
24/24 [==============================] - 16s 655ms/step - loss: 0.0445 - accuracy: 0.9905 - val_loss: 0.1474 - val_accuracy: 0.9511
Epoch 6/30
24/24 [==============================] - 15s 623ms/step - loss: 0.0260 - accuracy: 0.9892 - val_loss: 0.1103 - val_accuracy: 0.9620
Epoch 7/30
24/24 [==============================] - 15s 622ms/step - loss: 0.0153 - accuracy: 0.9946 - val_loss: 0.1243 - val_accuracy: 0.9783
Epoch 8/30
24/24 [==============================] - 15s 632ms/step - loss: 0.0268 - accuracy: 0.9905 - val_loss: 0.1297 - val_accuracy: 0.9620
Epoch 9/30
24/24 [==============================] - 15s 641ms/step - loss: 0.0257 - accuracy: 0.9932 - val_loss: 0.1127 - val_accuracy: 0.9783
Epoch 10/30
24/24 [==============================] - 16s 658ms/step - loss: 0.0280 - accuracy: 0.9919 - val_loss: 0.1220 - val_accuracy: 0.9728
Epoch 11/30
24/24 [==============================] - 16s 654ms/step - loss: 0.0223 - accuracy: 0.9905 - val_loss: 0.1165 - val_accuracy: 0.9511
Epoch 12/30
24/24 [==============================] - 16s 659ms/step - loss: 0.0185 - accuracy: 0.9946 - val_loss: 0.1481 - val_accuracy: 0.9728
Epoch 13/30
24/24 [==============================] - 16s 667ms/step - loss: 0.0162 - accuracy: 0.9946 - val_loss: 0.1151 - val_accuracy: 0.9728
Epoch 14/30
24/24 [==============================] - 16s 697ms/step - loss: 0.0151 - accuracy: 0.9973 - val_loss: 0.1423 - val_accuracy: 0.9783
Epoch 15/30
24/24 [==============================] - 16s 672ms/step - loss: 0.0129 - accuracy: 0.9973 - val_loss: 0.1280 - val_accuracy: 0.9728
```

FIG 6.1.7 TESTING

```
7/7 [==============================] - 3s 397ms/step - loss: 0.0911 - accuracy: 0.9617
[INFO] accuracy: 96.17%
[INFO] Loss: 0.09105636179447174
```

FIG 6.1.8 ACCURACY ON TESTING

Evaluating a machine learning model's performance using an independent dataset that it was not exposed to during training is the process of testing a model for correctness. This testing stage offers information on how well the model applies to fresh, untested data.Accuracy of the machine learning model, obtaining a thorough grasp of its efficacy and pinpointing possible areas for development. Remember that accuracy is only one indicator; for a more complex assessment of model performance, a comprehensive analysis utilising a variety of metrics is frequently advised.

ACCURACY

```
7/7 [==============================] - 3s 350ms/step - loss: 0.0876 - accuracy: 0.9856
[INFO] accuracy: 98.56%
[INFO] Loss: 0.08757727593183517
```

FIG 6.1.9 ACCURACY OF THE ALGORITHM

In machine learning, accuracy refers to how well a trained model performs on a different dataset that it was not exposed to during training. One frequent metric used to evaluate how well the model predicts the target accurately is accuracy.Compare the expected results of the model with the real results (ground truth) obtained from the testing dataset.

Accuracy=Number of Correct Predictions/Total Number of Predictions×100

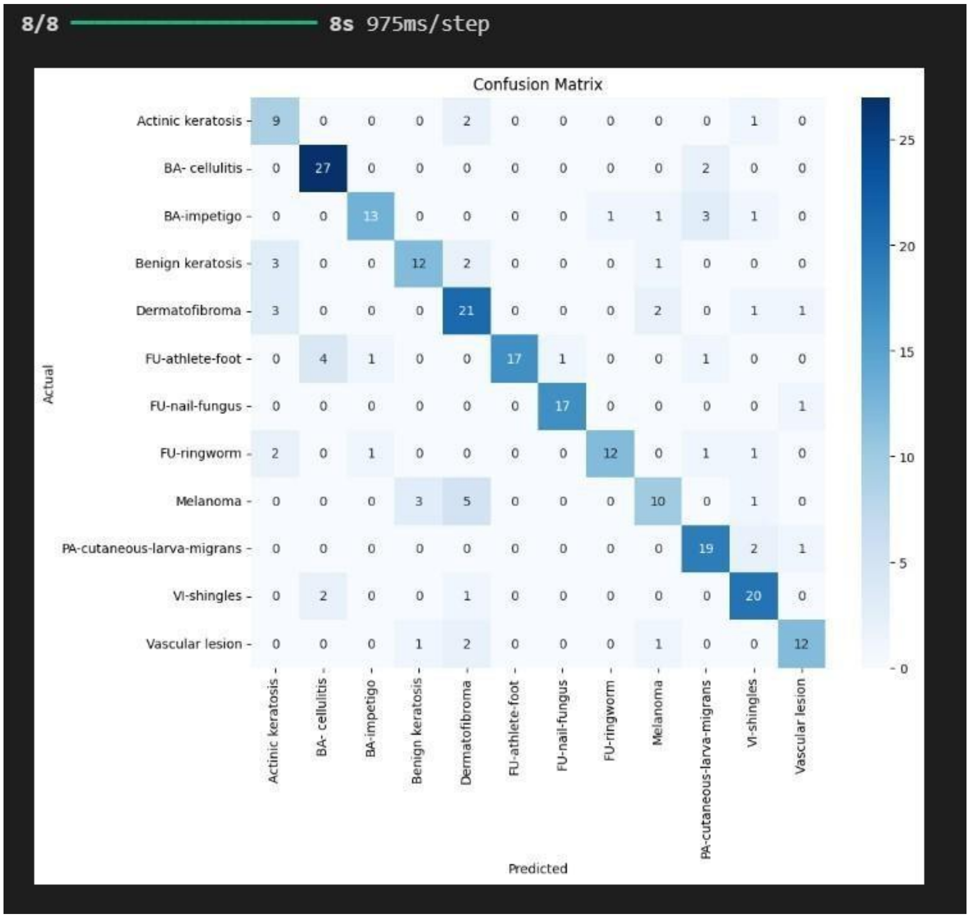

FIG 6.1.10 CONFUSION MATRIX

The model is trained and evaluated on subsets of the ground truth and then we evaluate the results looking for specific patterns.confusion matrix to provide more insights into the model's performance. It breaks down correct and incorrect predictions for each class, allowing you to identify false positives, false negatives, true positives, and true negatives.

6.2 RESULT & DISCUSSION

The skin disease prediction algorithm, trained on a diverse dataset, demonstrated accurate identification of common skin conditions, while Arduino-based health monitoring successfully measured blood pressure and oxygen levels. These individual components showcased promising results, laying the foundation for a comprehensive health system. Ongoing efforts will focus on seamless integration to create a unified platform, addressing challenges in data synchronization. Future work includes refining the user interface, implementing wireless connectivity, and incorporating robust security measures. Collaboration with healthcare professionals will ensure the system's alignment with medical standards, promising a reliable and user-friendly solution for skin disease prediction and health monitoring.

CHAPTER 7

CONCLUSION AND FUTURE WORK

APPENDICS

A1. SDG GOALS

Goal 3-Good Health and Well Being

HEALTHCARE PREVENTION

Preventive healthcare is a major component of SDG 3, and early identification and monitoring work to stop the spread of diseases and advance general well-being. This is the unifying subject of both programmes.

INNOVATION IN HEALTH TECHNOLOGY

The application of Skin disease and Arduino technology emphasises how innovation may advance health. This is consistent with the overarching goal of SDG 3, which is to use technology to improve healthcare.

EFFECT ON HEALTH WORLDWIDE

Potential to address health issues could have an international impact. This is consistent with the goal of SDG 3, which is to guarantee healthy lifestyles and well-being for all people globally.

A2.SOURCE CODE

ARDUINO

```
#include <Ethernet.h>

SoftwareSerial gps(2, 3);

// RX, TX
SoftwareSerial sim900a(7, 8);
// RX, TX for SIM900A

byte mac[] = {0x00, 0xAA, 0xBB, 0xCC,}
IPAddress ip(192,168,1,36);
EthernetServer server(80);

const int tempPin =A0;
const int alcohol = A3;
const int metalPin = 4;
const int heartPin = A1;

double alpha ;
double oldValue ;

void setup()
{
Ethernet.begin(mac, ip);
server.begin();
Serial.begin(9600);
gps.begin(9600);
sim900a.begin(9600);

pinMode(tempPin, INPUT);
pinMode(alcoholPin,
INPUT);pinMode(metalPin,
INPUT);
}
void loop() {
int pre = analogRead(A2);
int val = analogRead(tempPin);
float mv = (val / 1024.0) * 5000;
float cel = mv / 10;
int alcoholValue = analog(alcoholPin);
alcoholVal = alcoholVal * 5.0 / 1024;
```

```
int metalValue = digitalRead(metalPin);
int rawValue = analogRead(heartPin);
double value = alpha * oldValue + (1 - alpha) * rawValue;
value = value / 6.3;
processReadings(pre, cel, alcoholValue, metalValue, value);EthernetClient client =
server.available();

if (client)
{ handleClient(client, cel,pre);
}
}
void processReadings(int pre, float cel, int alcoholValue, int metalValue, double value)
{
Serial.println("-------- Sensor Readings-- ");
Serial.print("Pressure:");
Serial.println(pre);
Serial.print("Temperature:");
Serial.println(cel);
Serial.print("Alcohol:");
Serial.println(alcoholValue);
Serial.print("Metal: ");
Serial.println(metalValue);
Serial.print("Heart  Rate:");
Serial.println(value);
Serial.println("_________________");
}

void handleClient(EthernetClient client, float cel, int pre)
{
boolean currentLineIsBlank = true;
while (client.connected())
{
if (client.available())
{
 char c = client.read();
if (c == '\n' && currentLineIsBlank)
{
 sendHttpResponse(client, cel, pre);break;
}
if (c == '\n')
{
 currentLineIsBlank =true;
}
 else if (c != '\r')
```

```
{ currentLineIsBlank = false;
}
}
delay(1);
void sendHttpResponse(EthernetClient client, float cel, int pre)
{ client.println("HTTP/1.1 200 OK");
client.println("Content-Type:
text/html");
client.println("Connection:close");
client.println("Refresh:30");
client.println();
client.println("<html><head><meta        http-equiv='refresh'
content='30'></head><bodybgcolor=#FFFFFF>");
client.print("Temperature:");
client.print(cel);
client.print("Pressure:");
client.print(pre);
client.print("SMART-AMBULANCE");
client.println("</body></html>");
}

void readGPS() {
if (gps.available() > 0)
{
char c = gps.read();
 if (c== '$')
{
String y = gps.readStringUntil(',  ');
 if (y== "GPGGA")
}
}
}
}
```

SKIN DISEASE

App.py

```
from flask import render_template, jsonify, Flask, redirect, url_for, request
Import os
import io
import    numpy    as np
from PIL import Image
import keras.utils
from keras.model
 import model_from_json
```

```
app = Flask(_name_)

SKIN_CLASSES = {
0: 'Actinic Keratoses (Solar Keratoses) or intraepithelial Carcinoma (Bowen’s
disease)', 1:'Basal Cell Carcinoma',
2: 'Benign Keratosis',
3: 'Dermatofibroma',
4: 'Melanoma',
5: 'Melanocytic Nevi',
6:'Vascular skin
lesion'

}
@app.route('/')
def index():
return render_template('index.html')
@app.route('/signin')
def signin():
return render_template('signin.html')
@app.route('/signup')
def signup():
return render_template('signup.html')
@app.route('/dashboard',methods=['GET','POST'])
def dashboard():
return render_template('dashboard.html')
def findMedicine(pred):
ifpred == 0:
return "fluorouracil"
elifpred == 1:
return "Aldara"
elifpred == 2:
return "Prescription Hydrogen
Peroxide" elifpred == 3:
return "fluorouracil"
elifpred == 4:
return "fluorouracil (5-FU):"
elifpred == 5:
return "fluorouracil"
elifpred == 6:
return "fluorouracil"
@app.route('/detect', methods=['GET', 'POST']) defdetect():
```

```
json_response = {}
if request.method == 'POST':try:
file = request.files['file']
exceptKeyError:
return
make_response(jsonify({ 'error': 'No
file partin the request', 'code':'FILE',
'message': 'file is not valid'
}), 400)

imagePil = Image.open(io.BytesIO(file.read()))
# Savethe image to a BytesIO object
imageBytesIO = io.BytesIO()
imagePil.save(imageBytesIO,format='JPEG')
imageBytesIO.seek(0)
print("detected ")
 path = imageBytesIO

j_file = open('model.json', 'r')
loaded_json_model = j_file.read()
j_file.close()
model = model_from_json(loaded_json_model)
 model.load_weights('model.h5')
img = image.load_img(path,
target_size=(224, 224))img = np.array(img)
img = img.reshape((1,
224,224,3)) img = img/255
prediction = model.predict(img)
pred np.argmax(prediction)
 disease       =SKIN_CLASSES[pred]
accuracy = prediction[0][pred]
accuracy=round(accuracy*100,2)
medicine=findMedicine(pred)

json_response = {
"detected": False if pred == 2 else True, "disease":
disease,
"accuracy":   accuracy,  "medicine"  :
medicine, "img_path": file.filename,

}

return make_response(jsonify(json_response), 200)
```

```
else:
return render_template('detect.html')

if__name_=="_main_":
app.run(debug=True, port=3000)
}
}
}
}
```

Dashboard.css

```
* {

margin: 0;
padding: 0;
}
div img {
width: 350px;
position:
relative;bottom:
70px;
/* left: 500px; */
}

body
{
width:100%;

height: 100%;
position:relative;
background:#061A1F;
font-family:Cambria, Cochin, Georgia, Times, 'Times New Roman', serif;

/* background-color: red; */
display: flex;

flex-direction: column;
align-items: center;
justify-content:center;

    }
```

```
div img
{
 width: 350px;
 position:
 relative;bottom:
 70px;
 button {
 background-color: #58B2C9;
 border:none;
 border-radius:  8px;
 color:#FFF;
 font-size:    18px;
 padding:15px 51px;
 cursor: pointer;
 text-align: center;
  cursor: pointer;
  opacity: .8;

 }
 @media screen and (max-width:390px)
 {    div   img   {width:
 400px;
 position: relative;left: 0;
 }
 button {
 background-color: #58B2C9;
 flex-shrink: 0;
 border: none;
 color: white;
 padding:5px 22px;
 text- align:center;
 text-decoration: none;
 display:inline-block;
 font-size: 22px;
 border-radius: 12px;
 position: relative;
 bottom: 80px;
 left: 70px;
  cursor: pointer;}

 } */ button:hover
 {opacity: 1;
```

Detect.css

```
body
{ margin:
0;
padding: 0; position:
relative; min-height:
100vh;
}

.footer
{ position:
fixed; bottom:
0;
left: 0;
right: 0;
background-color:#fff;
padding: 20px;
}label {
color: #000;
font-size: 25px;
font-family:       Inter;
font-style: normal;
font-weight: 400;
line- height: normal;
}
```

Scan.css

```
body {
font-family: Arial, sans-
serif;display: flex;
flex-direction:
```

```
column;align-items:
center;justify-content:
center;height: 100vh;

  padding: 0;
}
h1 {
margin-bottom: 1rem;
}
video, canvas
{     max-width:
100%;
}
button {
margin-top:1rem;
padding: 0.5rem 1rem;
font-size: 1rem;
cursor: pointer;
}
```

Style.css

```
/* Hide scrollbar for all elements */
::-webkit-scrollbar
{ width: 0.5em;
}
::-webkit-scrollbar-track
{

Background
color:transparent;
}::-webkit-scrollbar-thumb
```

```
{ background
color:#03070
7;opacity: .9;
border-radius: 22px;
}
/* Optional: Style scrollbar on hover */
::-webkit-scrollbar-thumb:hover
{ background-color: #010303;
opacity: 1;
```

Script(1).js

```
const video = document.getElementById('video');
const canvas=document.getElementById('canvas');
const captureButton =document.getElementById('capture');
navigator.mediaDevices.getUserMedia({  video: true
})
  .then(stream =>
  { video.srcObject = stream;
  video.play();
  })
  .catch(err => {
  console.error('Error accessing the camera:', err);
  });
  // Capture the image when the button is
  clickedcaptureButton.addEventListener('click', () =>
  {const     context        =
  canvas.getContext('2d');
  canvas.width
```

```
video.videoWidth;
canvas.height
video.videoHeight;
context.drawImage(video, 0, 0, video.videoWidth, video.videoHeight);
});
```

Dashboard.html

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<linkrel="icon" type="image/x-icon" href="{{ url_for('static', filename='images/ICON1.ico') }}">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>DISKIN AI</title>
<link rel="stylesheet" href="/static/css/dashboard.css">
<link rel="stylesheet" href="/static/css/styles.css">
<body>
<div>
<img src="/static/images/ICON1.png">
</div>
<div>
<buttononclick="location.href='detect';"> Start detect</button>
</div></body>
</html>
```

Detected.html

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>Red Icon</title>
<link rel="icon" type="image/x-icon" href="{{ url_for('static', filename='images/ICON1.ico')
}}">
```

```
<link rel="stylesheet" type="text/css" href="/static/css/detected.css">
<link rel="stylesheet" href="/static/css/styles.css">
<link rel="stylesheet" type="text/css" href="/static/css/detect-form.css">
/head>
<body>
<div class="container">
<div class="image-container">
<img src="./red_icon-new.png" alt="Red Icon" class="colored-image">
</div>
<div class="text-container">
<p class="sorry-text">Sorry to say!</p>
<p class="disease-text">Disease has been detected.</p>
<button id="showFormButton">Click Here</button>
</div><div id="formContainer" class="center hidden">
<formaction="https://www.google.com/maps/search/Nahdi+Pharmacy/@21.5349688,39.2280073,14z/data=
!3m1!4b1?entry=ttu">
<label for="diseaseType">Type of disease</label><br>
<input id="diseaseType" type="text" placeholder="Enter the type of disease here"><br>
<label for="diseasePeriod">Period of disease spread</label><br>
<input id="diseasePeriod" type="text" placeholder="Enter the period of
disease spreadhere"><br>

<label for="medicine">Recommended medicine</label><br>
<input id="medicine" type="text" placeholder="Enter the recommended medicine here"><br>
<p class="medicine-text"> For Medicine places : </p>
<button id="btn" type="submit"> Check the Map </button>
</form>
</div>
```

```
</div>
<script>
VarshowFormButton=document.getElementById("showFormButton");
showFormButton.addEventListener("click", function() {
var formContainer
document.getElementById("formContainer");
formContainer.classList.remove("hidden");
var     textContainer
        document.querySelector(".textcontainer");
textContainer.style.display = "none";
});
</script>
</body>
</html>
```

scan.html

```
<!DOCTYPE html>
<html lang="en">
<head>

<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<link rel="icon" type="image/x-icon" href="{{ url_for('static', filename='images/ICON1.ico')
}}">
<title>scan</title>
<link rel="stylesheet" href="/static/css/scan.css">
</head>
<body>
<h1>Simple Body Scanning Example</h1>
<video id="video" autoplay></video>
```

```
<canvas id="canvas"></canvas>
<button id="capture">Capture Image</button>
<script src="script.js"></script>
</div>
</body>
</html>
```

TRAINING AND TESTING

```
import pathlib
import tensorflow as tf
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import os
import PIL
from tensorflow import keras
from tensorflow.keras import layers
from tensorflow.keras.models import Sequential
from keras.layers import Dense, Dropout, Flatten, Conv2D,
MaxPool2Dfrom tensorflow.keras.optimizers import Adam
import random
 from glob import glob
 import seaborn as sns
 from tensorflow.keras.losses import
 SparseCategoricalCrossentropyimport matplotlib.pyplot as plt
 import matplotlib.image as img
 import warnings
 warnings.filterwarnings('ignore')
 os.environ['TF_CPP_MIN_LOG_LEVEL'] = '3'
 tf.compat.v1.logging.set_verbosity(tf.compat.v1.logging.ERROR)

 data_dir_train = pathlib.Path("C:/Users/HP/skin disease
 prediction/Split_smol/train")
 data_dir_test = pathlib.Path("C:/Users/HP/skin disease
 prediction/Split_smol/val")image_count_train =
 len(list(data_dir_train.glob('*/*.jpg')))print(image_count_train)
 image_count_test =
 len(list(data_dir_test.glob('*/*.jpg')))
 print(image_count_test)
```

```
 train_ds
 tf.keras.preprocessing.image_dataset_from_directory(
 data_dir_train,
 validation_split=0.2,
 subset="training",
 seed=123,
 image_size=(img_height,
 img_width),
 batch_size=batch_size)

 val_ds =
 tf.keras.preprocessing.image_dataset_from_directory(
 data_dir_train,
validation_split=2,
subset="validationseed
=123,
image_size=(img_height,
img_width), batch_size=batch_size)

test_ds
tf.keras.preprocessing.image_dataset_from_directo
ry( data_dir_test,
validation_split=9,
subset="validation",
seed=123,
image_size=(img_height,
img_width), batch_size=batch_size)


class_names = train_ds.class_names
print(class_names)
num_classes=len(class_names)

for i in
range(num_classes):
plt.figure(figsize=(5,5))
plt.subplot(1,1,1)
image =
img.imread(str(list(data_dir_train.glob(class_names[i]+'/*.jpg'))[1
])) plt.title(class_names[i])
plt.imshow(image)
```

```
for image_batch, labels_batch in train_ds.take(1):
print(image_batch.shape)

print(labels_batch.shape)
AUTOTUNE = tf.data.experimental.AUTOTUNE
train_ds.cache().shuffle(1000).
prefetch(buffer_size=AUTOTUNE)
val_ds = val_ds.cache().prefetch(buffer_size=AUTOTUNE)
from tensorflow.keras.layers import RandomFlip, RandomRotation


data_augmentation =
keras.Sequential([
RandomFlip("horizontal"),
RandomRotation(0.1),
])
plt.figure(figsize=(10, 10))
for images, _ in train_ds.take(1):
for i in range(9):
augmented_images = data_augmentation(images)
ax = plt.subplot(3, 3, i + 1)
plt.imshow(augmented_images[0].numpy().astype("uint8"))
plt.axis("off")
num_classes =
len(class_names) total = 0
all_count = []
class_name = []
for i in range(num_classes):
count =
len(list(data_dir_train.glob(class_names[i]+'/*.jpg')))total
+= count
print("total training image count = {}
\n".format(total))print("..............................")
for i in range(num_classes):
count =
len(list(data_dir_train.glob(class_names[i]+'/*.jpg')))
print("Class name = ",class_names[i])
print("count            = ",count)
print("proportion = ",count/total)
print("

-------------------------
------------------------- ")
all_count.append(count)
```

```
  class_name.append(class_names[i])

temp_df = pd.DataFrame(list(zip(all_count, class_name)), columns = ['count',
'class_name'])sns.barplot(data=temp_df, y="count", x="class_name")
plt.xticks(rotation=90)
plt.show()
print("Number of training samples: ",
len(train_ds)) print("Number of validation
samples: ", len(val_ds))
print(train_ds.take(1).element_spec)  # print out the shape and data type of the dataset
print(train_ds.take(1)) # print out the first data sample
print(val_ds.take(1))
input_shape = (100, 125,
3)
num_classes = 16

model = Sequential()
model.add(Conv2D(32, kernel_size=(3, 3),activation='relu',padding =
'Same',input_shape=input_shape))model.add(Conv2D(32,kernel_size=(3, 3),
activation='relu',padding = 'Same',)) model.add(MaxPool2D(pool_size = (2, 2)))
model.add(Dropout(0.16))

model.add(Conv2D(32, kernel_size=(3, 3),activation='relu',padding
= 'Same')) model.add(Conv2D(32,kernel_size=(3, 3),
activation='relu',padding =
'Same',))model.add(MaxPool2D(pool_size = (2, 2)))
model.add(Dropout(0.20))

model.add(Conv2D(64, (3, 3), activation='relu',padding = 'same'))
model.add(Conv2D(64, (3, 3), activation='relu',padding = 'Same'))
model.add(MaxPool2D(pool_size=(2,
2))) model.add(Dropout(0.25))

model.add(Flatten())
model.add(Dense(256,
activation='relu'))
model.add(Dense(128,
activation='relu'))
model.add(Dropout(0.4))
model.add(Dense(num_classes,
```

```
    activation='softmax'))model.summar
    y()
    model.add(layers.Conv2D(256, (3, 3), activation='relu'))
    model.add(layers.MaxPooling2D((2, 2)))
    model.add(layers.Flatten())
    model.add(layers.Dense(1024,
    activation='relu'))
model.add(layers.Dense(num_classes, activation='softmax'))
optimizer = Adam(learning_rate=0.0001, beta_1=0.9, beta_2=0.999, epsilon=None, decay=0.0,
amsgrad=False)model.compile(optimizer = optimizer , loss = "categorical_crossentropy",
metrics=["accuracy"])
for class_name in os.listdir(dataset_path):
class_path = os.path.join(dataset_path, class_name)

 # Iterate through each image in the class
 for image_file in os.listdir(class_path):
 image_path = os.path.join(class_path, image_file)

# Load and preprocess the image
img = load_img(image_path, target_size=(width, height,channels))  # Set width and height as per your
modelimg_array = img_to_array(img)
img_array /= 255.0  # Normalize pixel values to [0, 1]

# Append the image data and label
images.append(img_array)
labels.append(class_name)  # Assuming class names are used
as labelsdatagen = ImageDataGenerator(
featurewise_center=False, # set input mean to 0 over the
dataset samplewise_center=False, # set each sample mean to 0
featurewise_std_normalization=False, # divide inputs by std of the dataset
samplewise_std_normalization=False, # divide each input by its std
zca_whitening=False, # apply ZCA whitening
rotation_range=10, # randomly rotate images in the range (degrees, 0 to 180)
zoom_range = 0.1, # Randomly zoom image

width_shift_range=0.1, # randomly shift images horizontally (fraction of total width)
height_shift_range=0.1, # randomly shift images vertically (fraction of total height)
horizontal_flip=False, # randomly flip images
vertical_flip=False)  # randomly flip images

datagen.fit(x_train)
# Convert lists to numpy arrays
```

```
 X np.array(images)
y =np.array(labels)

# Use label encoding or one-hot encoding for categorical labels if
needed# For example, using label encoding:
from sklearn.preprocessing import LabelEncoder

label_encoder = LabelEncoder()
y = label_encoder.fit_transform(y)

x_train = np.asarray(x_train_o['image'].tolist())
x_test = np.asarray(x_test_o['image'].tolist())

x_train_mean = np.mean(x_train)
x_train_std = np.std(x_train)

x_test_mean = np.mean(x_test)
x_test_std = np.std(x_test)

x_train = (x_train - x_train_mean)/x_train_std
x_test = (x_test - x_test_mean)/x_test_std
# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42) history =
model.fit(train_generator,epochs=epochs,validation_data=validation_generator)

# Evaluate the model
test_generator =
datagen.flow_from_directory(data_directory,
target_size=(input_shape[0],
input_shape[1]), batch_size=batch_size,
class_mode='categorical',
subset='validation',
 shuffle=False,
 classes=classes
)
```

```
  test_loss, test_accuracy =
  model.evaluate(test_generator)print(f'Test Accuracy:
  {test_accuracy}')

# Make predictions
y_true = test_generator.classes
y_pred_probs = model.predict(test_generator)
y_pred = np.argmax(y_pred_probs, axis=1)

# Print classification report and confusion matrix
print(classification_report(y_true, y_pred,
target_names=classes))print(confusion_matrix(y_true,y_pred))
```

A3.SCREENSHOTS

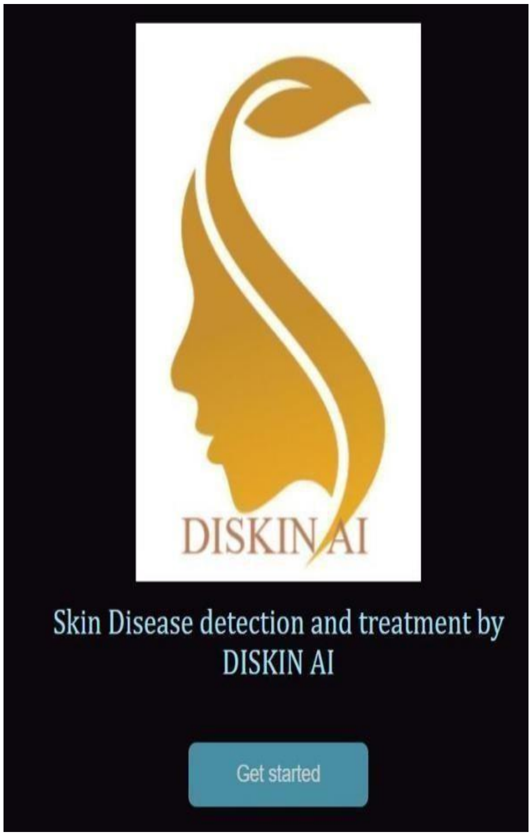

FIG A.3.1SCREENSHOT OF START PAGE

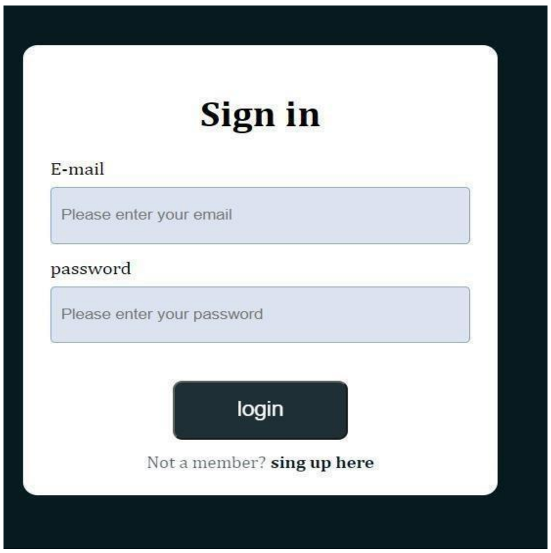

FIG A.3.2SCREENSHOT OF LOGIN PAGE

FIG A.3.3 SCREENSHOT OF UPLOADING PAGE

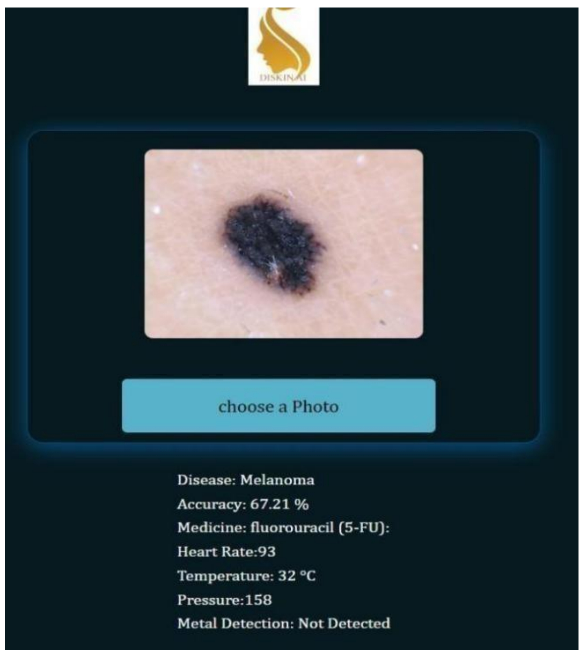

FIG A.3.4 SCREENSHOT OF RESULT PAGE

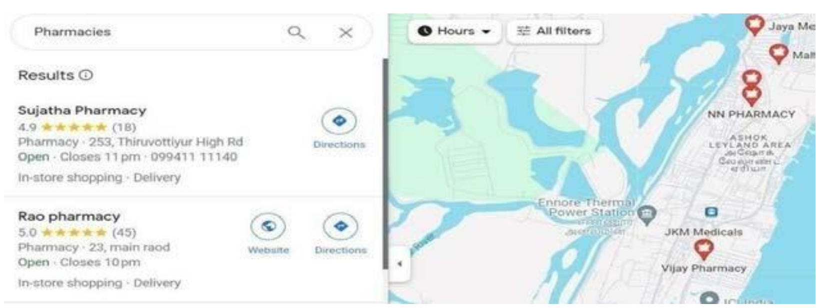

FIG A.3.5 SCREENSHOT OF CLINICS

A4.PLAGIARISM REPORT

VERSATILE AND REAL-TIME IOT BASED HEALTH MONITORING SYSTEM AND IDENTIFYING AILMENTS OF THE SKIN

1 Deepika P
Department of CSE,
Panimalar Engineering College,
Chennai, India.
deepikaprabakar22@gmail.com

1 Deepasree M
Department of CSE,
Panimalar Engineering College,
Chennai, India.
deepasreemari55@gmail.com

Indhumathi.V
Department of CSE,
Panimalar Engineering College
Chennai, India.
indhuindhuveera@gmail.com

2 Ms.V.Sathiya.
Associate Professor
2 epartment of CSE,
Panimalar Engineering college

Abstract---The Internet of Things (IOT) will be able to provide open access to selected subsets and seamlessly and transparently connect a wide range of diverse and heterogeneous end systems of information for the development of numerous digital services. The integrated system that combines an Arduino-based health monitoring module with skin disease detection component, the development of an integrated Skin Disease Prediction and Health Monitoring System utilizing Arduino and Flask, designed as a web application. The Health Monitoring component incorporates Arduino to gather real-time physiological data, including heart rate, body temperature, and blood pressure. This data is wirelessly transmitted to a central server and made accessible through a user-friendly web interface. a Flask-powered Skin Disease Prediction module employs a Convolutional Neural Network (CNN) algorithm to analyze uploaded images of skin anomalies. The CNN algorithm enhances the accuracy of skin disease classification, providing users with predictive insights. The cohesive platform not only empowers individuals to monitor their health in real-time but also offers an efficient and accurate tool for early detection and prediction of skin diseases, fostering a proactive approach to healthcare through an intuitive web interface.

Keywords: Internet of Things (IOT), Arduino, Flask ,metal sensor,Alcohol sensor,Temperature sensor,Heart rate sensor,Pressure sensor,GSM.

I. INTRODUCTION

IOT is a concept that not many people can define, but it's part of our present lives. IOT describes a vast global network of physical objects that are connected to the internet and are equipped to gather and share data.. As they can analyse data and employ AI to carry out specified tasks, these gadgets are frequently referred to as "smart devices". This paradigm surely finds utility in many domains, such as domestic automation, industrial automation, scientific aids, cellular smartphone healthcare, aged assistance, unique electrical electricity administration and actual grids, automotive, internet site on line web site traffic management, and many others .

IOT devices continuously capture and save information about users and their performance. Thus, it can be used to identify trends, difficulties, and potential issues and to offer recommendations in advance. This kingdom of affairs has led to the proliferation of specific and, sometimes, incompatible proposals for the accurate activity of IOT systems. Therefore, from a computer perspective, the middle of interest of an IOT network, mutually with the required backend close by selections and devices, on the different hand lacks a mounted incredible workout due to the truth of its novelty and complexity.

In the technical difficulties, the espouse of the IOT paradigm is in addition hindered by using capacity of the utilization of the lack of a clear and many times hooked up industrial business enterprise model that can enchantment to investments to promote the deployment of these utilized sciences . All our phones are IOT enabled. The moment we use Google Assistant , Amazon echo and Siri, we are using IOT.

II. LITERATURE SURVEY

Advancements in healthcare technology have spurred the development of integrated systems that combine health monitoring and skin disease detection. A comprehensive review of pertinent literature reveals key insights into the integration of various technologies, including the Internet of Things (IoT), machine learning, and image processing, aimed at enhancing healthcare diagnostics and monitoring..

The foundational work in health monitoring systems is exemplified by "Health Monitoring Based on Internet of Medical Things" [4], which elucidates the architecture and enabling technologies crucial for real-time health monitoring. This study underscores the significance of incorporating IoT principles into health monitoring systems, emphasizing the potential for continuous and remote health tracking. In conjunction, "IOT BASED HEALTH MONITORING SYSTEM" [8] introduces an IoT-centric health monitoring paradigm, delving into the integration of IoT for dynamic and personalized health assessment. These studies collectively lay the groundwork for the implementation of cutting-edge technologies in health monitoring.

Simultaneously, in the realm of dermatological diagnostics, the advent of deep learning is prominently featured in "Automatic Diagnosis of Skin Diseases Using Convolutional Neural Network" [13]. This research leverages convolutional neural networks (CNNs) to automate the diagnosis of skin diseases, showcasing the potential of artificial intelligence in streamlining dermatological assessments. Complementary to this, "A machine learning model for skin disease classification using convolution neural network" [14] advances the application of machine learning algorithms in skin disease classification, further substantiating the role of computational intelligence in dermatology.Here, we actually employ the traffic signals to control traffic without the aid of traffic officers. The mechanism for maintaining the traffic lighting

The fusion of these technologies is notably exemplified in "Skin Disease Classification from Image" [12], where advanced image processing techniques are employed to address challenges in image-based identification of skin diseases. This study contributes to the growing body of knowledge by presenting an innovative method for classifying skin diseases from images, highlighting the potential of image processing in enhancing diagnostic accuracy. Concurrently, "Advance study of skin diseases detection using image processing methods" [3] delves into the exploration of advanced image processing methods for skin disease detection, providing nuanced insights into the intricacies of image-based identification techniques.

In synthesis, these studies collectively underscore the transformative potential of integrated health monitoring systems and skin disease detection. The convergence of IoT, machine learning, and image processing technologies promises a paradigm shift in healthcare, offering more personalized, efficient, and accurate diagnostic and monitoring solutions. As these technologies continue to evolve, the envisioned integrated systems hold promise for revolutionizing healthcare practices.

A. INTEGRATED HEALTH MONITORING SYSTEM The evolution of health monitoring systems has seen significant contributions in recent literature. "Health Monitoring Based on Internet of Medical Things" by Kefeng Wei et al. [4] provides a foundational understanding of the architecture and technologies essential for robust health monitoring. Building on this, Prajoona Valsalan et al. [8] introduces an IoT-based health monitoring system, emphasizing the integration of diverse technologies for dynamic health assessment. Singh and Kaunert [10] delve into the integration of cutting-edge technologies, including IoT, to enhance health monitoring systems, reflecting the ongoing exploration of advanced technologies in healthcare.

B. SKIN DISEASE PREDICTION

In the domain of skin disease detection, machine learning and image processing have been pivotal. "Automatic Diagnosis of Skin Diseases Using Convolutional Neural Network" by T. Shanthi et al. [13] explores the application of CNNs for automated skin disease diagnosis, showcasing the potential of artificial intelligence in dermatology. Viswanatha Reddy Allugunti [14] presents a machine learning model for skin disease classification, emphasizing the role of convolutional neural networks in improving diagnostic accuracy. These studies collectively underscore the transformative impact of machine learning and image processing in dermatological diagnostic.

The ongoing evolution in health monitoring and skin disease detection, with a growing focus on integration and synergies between these domains. The collective insights from these studies pave the way for the development of advanced healthcare systems that offer a holistic view of individual well-being, blending internal health metrics with dermatological diagnostics for comprehensive healthcare management.

III. PROPOSED METHODOLOGY

In the rapidly evolving landscape of healthcare technology, the integration of real-time health monitoring and disease prediction systems presents a pioneering solution to enhance patient care and diagnostic capabilities. Our proposed system combines the robustness of an Arduino-based health monitoring module with the sophisticated capabilities of incorporating a Convolutional Neural Network (CNN) for skin disease prediction.This comprehensive approach aims to seamlessly provide continuous health insights, early anomaly detection, and accurate predictions of skin diseases, culminating in a versatile and proactive healthcare solution.

SKIN DISEASE PREDICTION

The skin disease prediction and health monitoring system aim to provide users with an efficient tool for early detection of skin conditions and continuous health monitoring.

FIG 3.1 ARCHITECTURE DIAGRAM

IMAGE

The dataset comprises a diverse set of high-quality images capturing different skin conditions. Each image is labeled with the corresponding skin disease for supervised learning tasks.To facilitate thorough analysis and feature extraction for machine learning, all images have high-quality pixels.

PREPROCESSING

Machine learning models to predict skin diseases from picture datasets, preprocessing is an essential step in the process. This initial stage consists of a number of procedures intended to improve the relevance and quality of the data, making it suitable for training reliable and accurate models. picture normalisation, scaling, and feature extraction are common preparation procedures used in the context of skin disease picture datasets. By standardising pixel values, normalisation ensures uniformity across images and guards against biases in the training of models. In order to preserve consistency in the dimensions of images and enable effective computing during model training, resizing is essential.

```
Found 924 files belonging to 8 classes.
Using 740 files for training.
```

FIG 3.2 Preprocessing

ARDUINO

This platform's central component, the Arduino board, has a microcontroller that carries out preprogrammed instructions. Arduino boards each designed to meet particular requirements. They havemicrocontrollers, USB ports, and a variety of input/output (I/O) pins. Together, these parts give users the ability to connect actuators, sensors, and other electronic devices, which makes the Arduinoa flexible platform for creative expression.

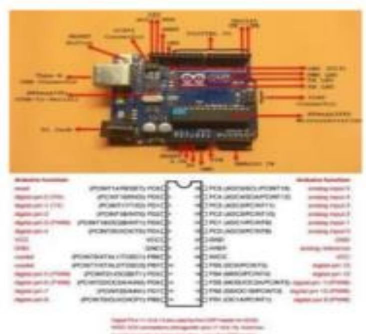

Fig 3.3 Arduino Board

SENSORS

The sensors that measure blood pressure, temperature, heart rate, metal detector, and alcohol detector are linked to the Arduino board.

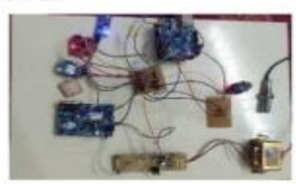

Fig 3.4 Arduino setup

FLASK

Flask is a lightweight and flexible web framework for Python that is commonly used for building web applications.The Flask application for health monitoring and skin disease prediction aims to provide users with a seamless and user-friendly platform to track their health metrics and receive predictions related to skin conditions. Leveraging Flask's capabilities, the application offers a range of features including user authentication, data visualization

HTTP requests. Views are implemented using Python functions, allowing developers to define the logic for processing requests and generating responses.Flask-Login extension facilitates user authentication and session management. It enables features such as user registration, login, logout, and password reset, ensuring secure access to the application.Flask integrates with libraries such as Plotly or Matplotlib to visualize health metrics data in the form of charts and graphs. These visualizations aid users in understanding their health trends and patterns effectively.Flask serves as the backend for integrating machine learning models trained on skin disease datasets. Users can submit images or descriptions of skin conditions, and Flask handles the prediction process, providing relevant results to the user.Flask provides mechanisms for error handling and logging, ensuring robustness and monitoring of the application. Custom error handlers can be implemented to gracefully handle exceptions and provide meaningful error messages to users.The Flask application for health monitoring and skin disease prediction demonstrates the effectiveness of Flask in building scalable and feature-rich applications catering to the healthcare domain.Flask is suitable for the backend of a skin disease prediction application, the frontend (HTML, CSS, and JavaScript) and the machine learning model itself are crucial components that should be developed and integrated accordingly. Additionally, ensure that the application complies with relevant data privacy and security standards, especially when dealing with sensitive medical information.

IV. PERFORMANCE ANNALYSIS

TRAINING

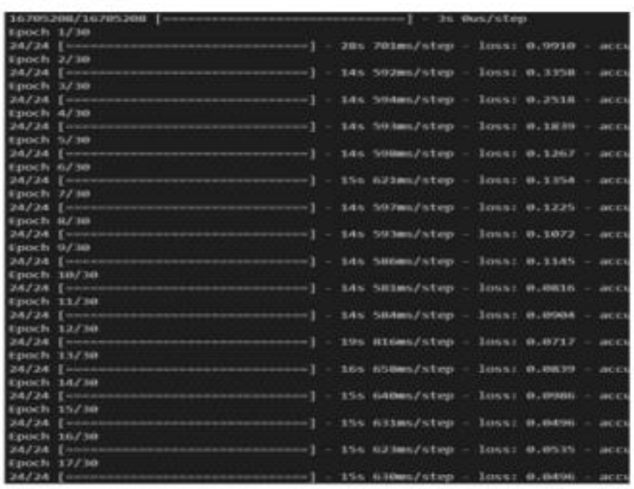

FIG 4.1 TRAINING ACCURACY

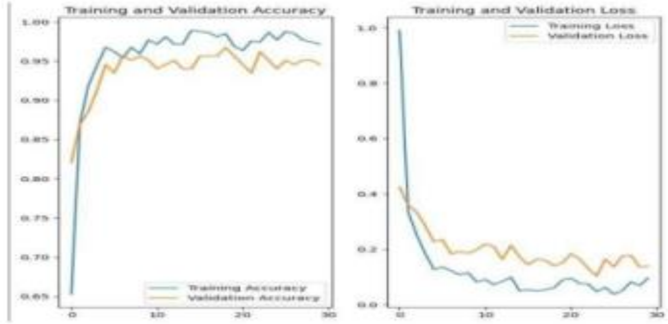

Fig 4.2 Line graph of training

TESTING

Fig 4.3 Testing of Accuracy

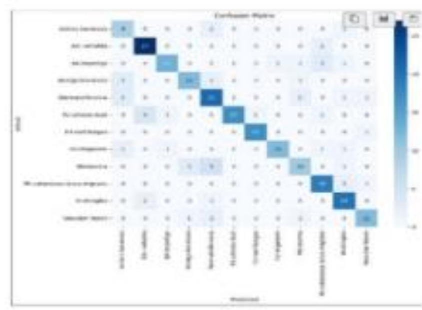

Fig 4.4 Confusion matrix

ACCURACY

Fig 4.5 Accuracy of algorithm

In machine learning, accuracy refers to how well a trained model performs on a different dataset that it was not exposed to during training. One frequent metric used to evaluate how well the model predicts the target accurately is accuracy.Compare the expected results of the model with the real results (ground truth) obtained from the testing dataset.

Accuracy=Number of Correct Predictions/Total Number of Prediction ×100

IV. RESULT AND DISCUUSION

The integrated health monitoring and skin disease prediction system showed strong performance and extensive insights into individual wellbeing. With the help of Arduino, the health monitoring system was able to efficiently track important metrics including blood pressure, temperature, and heart rate in real time. Remote monitoring was made possible by the GSM technology's smooth wireless connectivity. Reliableand extended functioning was ensured by power- efficient tactics and ongoing monitoring systems. Thorough testing confirmed the system's accuracy and dependability in a variety of settings, and privacy and ethical policies complied with accepted norms. Concurrently, skin illness prediction system demonstrated sophisticated image processing ability to identify possible skin conditions. Accurate disease detection was proved by the high-resolution camera's ability to capture skin images and the image processing techniques used.

Fig 5.1 Screenshot Of Start Page

Fig 5.2 Screenshot Of Login Page

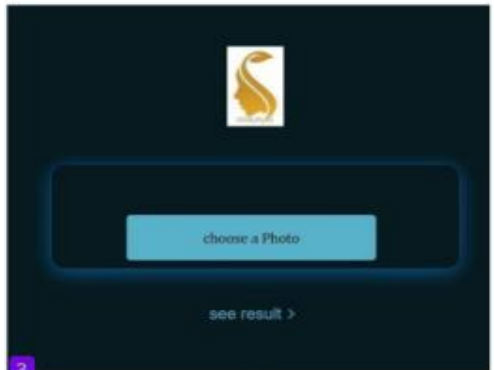

Fig 5.3 Screenshot Of Uploading Image

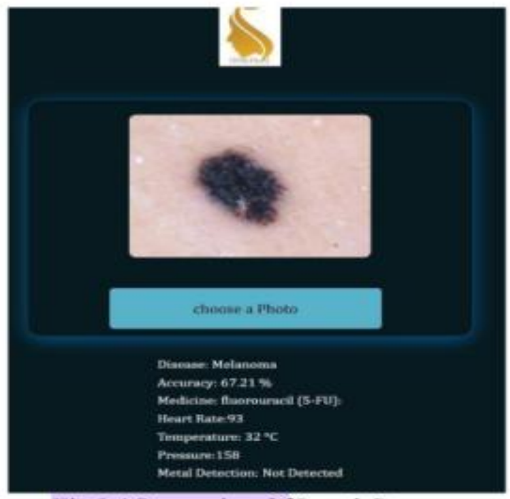

Fig 5.4 Screenshot Of Result Page

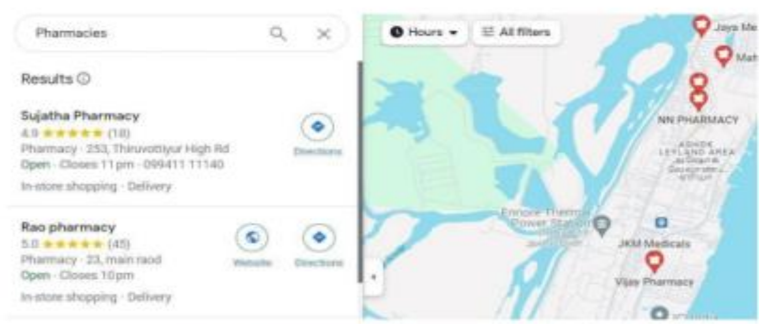

Fig 5.5 Screenshot Of Clinics

V. CONCLUSION

The collaborative Arduino and skin disease prediction technologies in the proposed health monitoring and skin disease prediction system offers a promising and comprehensive solution for personalized healthcare. The Arduino-based health monitoring module demonstrates real-time monitoring of vital parameters with precision, while the skin disease prediction component effectively employs advanced image processing for accurate identification of potential skin issues. This unified approach provides users with a holistic overview of their well-being, combining internal health parameters and external skin condition analysis. Rigorous testing ensures the system's reliability, ethical considerations are met, and user-friendly documentation facilitates seamless adoption.

VI. FUTURE ENHANCEMENT

Multimodal strategy is used to improve the capabilities and user experience of the integrated health monitoring and skin disease prediction system. For a more comprehensive assessment of health, the health monitoring system must be advanced by monitoring factors such as glucose levels, oxygen saturation, and other pertinent health metrics. Predictive analytics using machine learning algorithms can help the system anticipate possible health problems and suggest preventative actions. Furthermore, consumers will have easy access to current health information by improving the user experience through the creation of a mobile application.

In the field of skin disease prediction will focus on increasing the system's usefulness and improving its diagnostic accuracy. Increasing the number of skin picture entries in the database will help the model identify a wider range of skin problems. More precise and sophisticated predictions are promised when cutting-edge artificial intelligence systems are integrated with picture analysis. Working together with dermatologists and other medical specialists will help to further validate and improve the technology for use in clinical settings. Ensuring compatibility with portable skin inspection equipment encourages increased personal healthcare engagement by empowering users to check themselves.

REFERENCES

[1] Abderahman Rejeb , Karim Rejeb , Horst Treiblmaier , Andrea Appolloni , Salem Alghamdi , Yaser Alhasawi , Mohammad Iranmanesh "The Internet of Things (IoT) in healthcare"Internet of Things Volume 22, July 2023, 100721

[2] AlDera S.A., Othman M.T.B.A model for classification and diagnosis of skin disease using machine learning and image processing techniques Science Direct (2019)

[3] Jagdisetal.,CruzVargas J.A.D.L., Camacho M.E. R.Advance study of skin diseases detection using image processing methods NVEO(2022)

[4] Kefeng Wei, Lincong Zhang , Yi Guo, And Xin Jiang Health Monitoring Based on Internet of Medical Things: Architecture, Enabling Technologies, and Applications Feb 13, 2020

[5] Kushal Pokhrel,Cesar Sanin,Md. Kowsar Hossain Sakib,Md Rafiqul Islam Improved Skin Disease Classification with Mask R-CNN and Augmented Dataset Research Gate(2023)

[6] Kinza Shafique, "Internet of things (IoT) for next-generation smart systems: A review of current challenges, future trends and prospects for emerging 5G-IoT scenarios" Ieee Access, 8 (2020).

[7] N. Gessert, T. Sentker, F. Madesta, R. Schmitz, H. Kniep, I. Baltruschat,5 Werner and A. Schlaefer, "Skin Lesion Classification Using CNNs with Patch-Based Attention and Diagnosis-Guided Loss Weighting," IEEE Transactions on Biomedical Engineering, 2019

[8] Prajoona Valsalan, Tariq Ahmed Barham Baomar, Ali Hussain Omar Baabood "IOT BASED HEALTH MONITORINGSYSTEM", Research Gate 2020

[9] R. Lederman, O. Ben-Assuli, T.H. Vo "The role of the Internet of Things in Healthcare in supporting clinicians and patients: a narrative review"Health Policy Technol., 10 (2021).

[10] Singh, B., & Kaunert, C. (2024). Integration of Cutting-Edge Technologies such as Internet of Things (IoT) and 5G in Health Monitoring Systems GLS Law

Journal , 6(1), 13 - 20.

[11] Steven Hayward , Katherine van Lopik, Andrew West."A holistic approach to health and safety monitoring: Framework and technology perspective".Internet of Things 20 (2022) 100606

[12] Tanvi Goswami ,Vipul K. Dabhi ,Harshadkumar B. Prajapati"Skin Disease Classification from Image "2020 6th International Conference on Advanced Computing & Communication Systems (ICACCS)

[13] T. Shanthi, R.S. Sabeenian, R. Anand, "Automatic Diagnosis of Skin Diseases Using Convolutional Neural Network", Elsevier, Microprocessors and Microsystems volume 76, July2020, 103074

[14] Viswanatha Reddy Allugunti "A machine learning model for skin disease classification using convolution neural network" International Journal of Computing, Programming and Database Management 2022; 3(1): 141-147.

[15] Y.A. Qadri, A. Nauman, Y.B. Zikria, A.V. Vasilakos, S.W. Kim"The future of healthcare Internet of Things: a survey of emerging technologies"IEEE Commun. Surv. Tutor., 22 (2020), pp. 1121-1167.

[16] D. Sharathchandra, M. Raghu Ram,ML Based Interactive Disease Prediction Model,20 APRIL2022 IEEE Delhi Section Conference (DELCON).

[17] Wsandhyarani,Boddupelli Durgabhavani et.al,Classification of Skin Disease using CNN,21October 2023 International Conference on Evolutionary Algorithms and Soft Computing Techniques (EASCT)

[18] X. He, Y. Wang, S. Zhao, and C. Yao, "Deep metric attention learning for skin lesion classification in dermoscopy images," Complex and Intelligent Systems, vol. 8, no. 2, pp. 1487–1504, Apr. 2022, doi: 10.1007/S40747-021-00587-4/FIGURES/8.

[19] V. Rawat, D. P. Singh, N. Singh, P. Kumar, and T. Goyal, "A Comparative Study of various Skin Cancer using Deep Learning Techniques," Proceedings of International Conferenc on Computational Intelligence and Sustainable Engineering Solution, CISES 2022, pp. 505–511,2022, doi: 10.1109/CISES54857.2022.9844409.

[20] V. B. Kumar, S. S. Kumar, and V. Saboo, "Dermatological Disease Detection Using Image Processing and Machine Learning," 2016 3rd International Conference on Artificial Intelligence and Pattern Recognition, AIPR 2016, pp. 88–93, Oct. 2016, doi: 10.1109/ICAIPR.2016.7585217.

RE-2022-219744-plag-report

ORIGINALITY REPORT

5%	1%	4%	2%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	"Smart Intelligent Computing and Communication Technology", IOS Press, 2021 Publication	1%
2	"Intelligent Systems and Computer Technology", IOS Press, 2020 Publication	1%
3	Submitted to University of Luton Student Paper	1%
4	hdl.handle.net Internet Source	<1%
5	www.atlantis-press.com Internet Source	<1%
6	Submitted to University of Hertfordshire Student Paper	<1%
7	www.mdpi.com Internet Source	<1%
8	Tejas S Koundinya, Sathwik C Gowda, Jyoti R Munavalli. "Dhanvantari: An intelligent diagnosis tool to classify malignant skin	<1%

REFERENCES

[1] Abderahman Rejeb , Karim Rejeb , Horst Treiblmaier , Andrea Appolloni , Salem Alghamdi , Yaser Alhasawi , Mohammad Iranmanesh "The Internet of Things (IoT) in healthcare"Internet of Things Volume 22, July 2023, 100721

[2] AlDera S.A., Othman M.T.B.A model for classification and diagnosis of skin disease using machine learning and image processing techniques Science Direct (2019)

[3] Jagdisetal.,CruzVargas J.A.D.L., Camacho M.E.R.Advance study of skin diseases detection using image processing methods NVEO(2022)

[4] Kefeng Wei, Lincong Zhang , Yi Guo, And Xin Jiang Health Monitoring Based on Internet of Medical Things: Architecture, Enabling Technologies, and Applications Feb 13, 2020

[5] Kushal Pokhrel,Cesar Sanin,Md. Kowsar Hossain Sakib,Md Rafiqul Islam Improved Skin Disease Classification with Mask R-CNN and Augmented Dataset Research Gate(2023)

[6] Kinza Shafique, "Internet of things (IoT) for next-generation smart systems: A review of current challenges, future trends and prospects for emerging 5G-IoT scenarios" Ieee Access, 8 (2020).

[7] N. Gessert, T. Sentker, F. Madesta, R. Schmitz, H. Kniep, I. Baltruschat,5 Werner and A. Schlaefer, "Skin Lesion Classification Using CNNs with Patch-Based Attention and Diagnosis-
Guided Loss Weighting," IEEE Transactions on Biomedical Engineering, 2019

[8] Prajoona Valsalan, Tariq Ahmed Barham Baomar, Ali Hussain Omar Baabood "IOT BASED HEALTH MONITORINGSYSTEM", Research Gate 2020

[9] R. Lederman, O. Ben-Assuli, T.H. Vo "The role of the Internet of Things in Healthcare in supporting clinicians and patients: a narrative review"Health Policy Technol., 10 (2021).

[10] Singh, B., & Kaunert, C. (2024). Integration of Cutting-Edge Technologies such as Internet of Things (IoT) and 5G in Health Monitoring Systems GLS Law Journal , 6(1), 13 - 20.

[11] Steven Hayward , Katherine van Lopik, Andrew West."A holistic approach to health and safety monitoring: Framework and technology perspective".Internet of Things 20 (2022) 100606

[12] Tanvi Goswami ,Vipul K. Dabhi ,Harshadkumar B. Prajapati"Skin Disease Classification from Image "2020 6th International Conference on Advanced Computing & Communication Systems (ICACCS)

[13] T. Shanthi, R.S. Sabeenian, R. Anand, "Automatic Diagnosis of Skin Diseases Using Convolutional Neural Network", Elsevier, Microprocessors and Microsystems volume 76, July 2020, 103074

[14] Viswanatha Reddy Allugunti "A machine learning model for skin disease classification using convolution neural network" International Journal of Computing, Programming and Database Management 2022; 3(1): 141-147.

[15] Y.A. Qadri, A. Nauman, Y.B. Zikria, A.V. Vasilakos, S.W. Kim"The future of healthcare Internet of Things: a survey of emerging technologies"IEEE Commun. Surv. Tutor., 22 (2020), pp. 1121-1167.

[16] D. Sharathchandra, M. Raghu Ram,ML Based Interactive Disease Prediction

Model,20 APRIL 2022 IEEE Delhi Section Conference (DELCON).

[17] Wsandhyarani,Boddupelli Durgabhavani et.al,Classification of Skin Disease using CNN,21 October 2023 International Conference on Evolutionary Algorithms and Soft Computing Techniques (EASCT)

[18] X. He, Y. Wang, S. Zhao, and C. Yao, "Deep metric attention learning for skin lesion classification in dermoscopy images," Complex and Intelligent Systems, vol. 8, no. 2, pp. 1487–1504, Apr. 2022, doi: 10.1007/S40747-021-00587-4/FIGURES/8.

[19] V. Rawat, D. P. Singh, N. Singh, P. Kumar, and T. Goyal, "A Comparative Study of various Skin Cancer using Deep Learning Techniques," Proceedings of International Conferenceon Computational Intelligence and Sustainable Engineering Solution, CISES 2022, pp. 505–511,2022,doi: 10.1109/CISES54857.2022.9844409.

[20] V. B. Kumar, S. S. Kumar, and V. Saboo, "Dermatological Disease Detection Using Image Processing and Machine Learning," 2016 3rd International Conference on Artificial Intelligence and Pattern Recognition, AIPR 2016, pp. 88–93, Oct. 2016, doi: 10.1109/ICAIPR.2016.7585217