A Project report

On

NEURAL NETWORK BASED VITAMIN DEFICIENCY DETECTION USING IMAGES

Submitted in partial fulfillment of the requirements

for the award of the degree of

BACHELOR OF TECHNOLOGY

in

Computer Science & Engineering

by

R. PALLAVI	**204G1A0569**
M. SAI KIRAN	**204G1A0586**
C. VAMSI	**214G5A0512**
G. SREEKAR VAMSI KRISHNA	**204G1A05A1**

Under the Guidance of

Mr. P. Shajahan M.Tech

Assistant Professor

Department of Computer Science & Engineering

SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY

(AUTONOMOUS)

(Affiliated to JNTUA, Accredited by NAAC with 'A' Grade, Approved by AICTE, New Delhi & Accredited by NBA (EEE, ECE & CSE)

Rotarypuram Village, BK Samudram Mandal, Ananthapuramu-515701

2023-2024

SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY

(AUTONOMOUS)

(Affiliated to JNTUA, Accredited by NAAC with 'A' Grade, Approved by AICTE, New Delhi & Accredited by NBA (EEE, ECE & CSE)

Rotarypuram Village, BK Samudram Mandal, Ananthapuramu-515701

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Certificate

This is to certify that the Project report entitled **NEURAL NETWORK BASED VITAMIN DEFICIENCY DETECTION USING IMAGES** is the bonafide work carried out by **R. Pallavi, M. Sai Kiran, C. Vamsi, G. Sreekar Vamsi Krishna** bearing Roll Number **204G1A0569, 204G1A0586, 214G5A0512, 204G1A05A1** in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science & Engineering** during the academic year 2023 - 2024.

Project Guide

Mr. P. Shajahan M.Tech.
M.Tech.,(Ph.D.)Assistant Professor

Head of the Department

Mr. P. Veera Prakash
Assistant Professor & HOD

Date:

EXTERNAL EXAMINER

Place: Rotarypuram

DECLARATION

We, Ms. R. Pallavi with reg no: 204G1A0569 , Mr. M. Sai Kiran with reg no: 204G1A0586, Mr. C. Vamsi with reg no: 214G5A0512 , Mr. G. Sreekar Vamsi Krishna with reg no: 204G1A05A1 students of SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY , Rotarypuram , hereby declare that the dissertation entitled **"NEURAL NETWORK BASED VITAMIN DEFICIENCY DETECTION USING IMAGES"** embodies the report of our project work carried out by us during IV year Bachelor of Technology under the guidance of **Mr. P. Shajahan** **M.Tech**. Department of CSE, **SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY**, and this work has beensubmitted for the partial fulfillment of the requirements for the award of the Bachelor of Technology degree.

The results embodied in this project have not been submitted to any other University of Institute for the award of any Degree or Diploma.

R. PALLAVI	Reg no: 204G1A0569
M. SAI KIRAN	Reg no: 204G1A0586
C. VAMSI	Reg no: 214G5A0512
G. SREEKAR VAMSI KRISHNA	Reg no: 204G1A05A1

Vision & Mission of the SRIT

Vision:

To become a premier Educational Institution in India offering the best teaching and learning environment for our students that will enable them to become complete individuals with professional competency, human touch, ethical values, service motto, and a strong sense of responsibility towards environment and society at large.

Mission:

- Continually enhance the quality of physical infrastructure and human resources to evolve in to a center of excellence in engineering education.

- Provide comprehensive learning experiences that are conducive for the students to acquire professional competences, ethical values, life-long learning abilities and understanding of the technology, environment and society.

- Strengthen industry institute interactions to enable the students work on realistic problems and acquire the ability to face the ever changing requirements of the industry.

- Continually enhance the quality of the relationship between students and faculty which is a key to the development of an exciting and rewarding learning environment in the college.

Vision & Mission of the Department of CSE

Vision:

To evolve as a leading department by offering best comprehensive teaching and learning practices for students to be self-competent technocrats with professional ethics and social responsibilities.

Mission:

DM 1: Continuous enhancement of the teaching-learning practices to gain profound knowledge in theoretical & practical aspects of computer science applications.

DM 2: Administer training on emerging technologies and motivate the students to inculcate self-learning abilities, ethical values and social consciousness to become competent professionals.

DM 3: Perpetual elevation of Industry-Institute interactions to facilitate the students to work on real-time problems to serve the needs of the society.

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of anytask would be incomplete without the mention of people who made it possible, whose constant guidance and encouragement crowned our efforts with success. It is a pleasantaspect that we have now the opportunity to express our gratitude for all of them.

It is with immense pleasure that we would like to express our indebted gratitude to our Guide **Mr. P. Shajahan, M.Tech., Assistant Professor, Computer Science & Engineering**, who has guided us a lot and encouraged us in every step of the project work. We thank him for the stimulating guidance, constant encouragement and constructive criticism which have made possible to bring out this project work.

We are very much thankful to **Mr. P. Veera Prakash, M.Tech., (Ph.D.), Assistant Professor & Head of the Department, Computer Science & Engineering,** for his kind support and for providing necessary facilities to carry out the work.

We wish to convey our special thanks to **Dr. G. Bala Krishna, Ph.D., Principal** of **Srinivasa Ramanujan Institute of Technology** for giving the required information in doing our project work. Not to forget, we thank all other faculty and non- teaching staff, and our friends who had directly or indirectly helped and supported us in completing our project in time.

We also express our sincere thanks to the Management for providing excellent facilities**.**

Finally, we wish to convey our gratitude to our families who fostered all the requirements and facilities that we need.

Project Associates

204G1A0569 PALLAVI R

204G1A0586 SAI KIRAN M

214G5A0512 VAMSI C

204G1A05A1 SREEKAR VAMSI KRISHNA G

.

ABSTRACT

Vitamin deficiencies pose a significant global health concern, resulting in various debilitating conditions. Detecting these deficiencies early on is crucial for effective intervention and prevention. This study presents a novel technique that uses image processing and neural networks to identify vitamin deficiencies. The system utilizes image analysis techniques to evaluate physical signs of deficiencies, such as abnormalities in the skin, nails, or eyes. Detailed images of these symptoms are collected and processed to extract relevant features. A convolutional neural network (CNN) is then utilized to classify these features and identify potential deficiencies. The neural network is trained on a comprehensive dataset of images representing different vitamin deficiencies, ensuring accurate and robust detection capabilities. To further enhance the system's performance, transfer learning and data augmentation techniques are employed. Experimental results demonstrate the system's effectiveness in accurately detecting vitamin deficiencies. Additionally, the non-invasive nature of image-based detection makes this approach convenient and accessible for widespread screening and monitoring, especially in resource-limited settings. In conclusion, this proposed methodology offers a promising solution for early detection and intervention of vitamin deficiencies, contributing to improved public health outcomes and overall well-being. This innovative approach bridges the gap between technology and healthcare, potentially revolutionizing the global fight against vitamin deficiencies.

Keywords: Vitamin deficiency, deep learning, CNN, OpenCV

CONTENTS

		Page No.
List of Figures		**VIII**
List of Abbreviations		**IX**
Chapter 1	**Introduction**	**1**
	1.1 Problem Statement	3
	1.2 Objectives	3
	1.3 Scope of Project	3
Chapter 2	**Literature Survey**	**4**
Chapter 3	**Methodology**	**7**
	3.1 Machine Learning	7
	3.2 Algorithm Used	9
Chapter 4	**System Requirements**	**13**
	4.1 Functional Requirements	13
	4.2 Non-Functional Requirements	13
	4.3 Python Libraries	14
	4.4 Software Requirements	30
	4.5 Hardware Requirements	30
Chapter 5	**System Analysis and Design**	**31**
	5.1 UML Diagrams	31
	5.2 System Architecture	37
	5.3 Flow Chart	39

Chapter 6	**Implementation**	**40**
	6.1 Datasets	42
	6.2 Data Pre-Processing	43
	6.3 Data Training	44
	6.4 Data Classification	44
Chapter 7	**Testing**	**45**
	7.1 Functionality Testing	45
	7.2 Usability Testing	46
	7.3 Interface Testing	46
	7.4 Performance Testing	46
	7.5 Unit Testing	47
	7.6 Integration Testing	47
	7.7 Acceptance Testing	48
	7.8 System Testing	48
	7.9 White Box Testing	49
	7.10 Black Box Testing	49
Chapter 8	**Results**	50
Conclusion		51
References		52
Publication Paper		54
Publication Certificates		60

List of Figures

Fig No.	Description	PageNo
Fig.3.1	Types of Machine Learning	7
Fig.3.2	Process of Supervised Learning	8
Fig.3.3	Process of Unsupervised Learning	8
Fig.3.4	Reinforcement Learning	9
Fig.3.5	Convolution Operation	10
Fig.3.6	Feature Maps	10
Fig.3.7	Image Scan	11
Fig.3.8	CNN Architecture	12
Fig.4.1	Python Installation	14
Fig.4.2	Pip install Numpy	15
Fig.5.1	Use Case Diagram	31
Fig.5.2	Class Diagram	32
Fig.5.3	Sequence Diagram	33
Fig.5.4	Collaboration Diagram	33
Fig.5.5	Activity Diagram	34
Fig.5.6	System Architecture	36
Fig.5.7	Flowchart of the system	37
Fig.6.1	Block diagram of System	38
Fig.6.2	Dataset Collection	41
Fig.8.1	Login Page	50
Fig.8.2	Registration Page	50
Fig.8.3	Home Page	51
Fig.8.4	Upload Page	51
Fig.8.5	Result Page	52

LIST OF ABBREVIATIONS

DFD	Data Flow Diagram
CNN	Convolution Neural Network
ER	Entity Relationship
SRS	System Requirements Specification
UML	Unified Modelling Language

CHAPTER 1

INTRODUCTION

Healthcare Need: Vitamin deficiencies can lead to serious health problems if not detected and treated early. However, traditional detection methods might not be readily accessible or efficient.

Technological Advancements: The rapid development in machine learning and deep learning offers an opportunity to revolutionize healthcare diagnostics.

Accessibility and Affordability: There is a growing need for affordable, non-invasive, and accessible healthcare solutions, especially in resource-limited settings.

Data-Driven Healthcare: The increasing availability of health-related data and the ability to process it efficiently can lead to more personalized and preventive healthcare strategies.

Global Health Impact: Addressing vitamin deficiencies effectively can have a significant positive impact on global health, reducing the burden of disease related to malnutrition.

Develop a Non-Invasive Detection System: Utilizing machine learning and deep learning algorithms to analyze indicators of vitamin deficiency from easily obtainable data like dietary patterns, physical symptoms, etc.

Improve Accuracy and Speed of Diagnosis: To provide a faster and more accurate diagnosis than conventional methods.

Increase Accessibility: Making vitamin deficiency detection more accessible to underprivileged and remote areas.

Contribute to Preventive Healthcare: To aid in early intervention and preventive healthcare measures.

Vitamin deficiency is the condition of a long-term lack of a vitamin. When caused by not enough vitamin intake it is classified as a primary deficiency, whereas when due to an underlying disorder such as malabsorption it is called a secondary deficiency.

An underlying disorder may be metabolic – as in a genetic defect for converting tryptophan to niacin or from lifestyle choices that increase vitamin needs, such as smoking or drinking alcohol. Government guidelines on vitamin deficiencies advise certain intakes for healthy people, with specific values for women, men, babies, the elderly, and during pregnancy or breastfeeding. Many countries have mandated vitamin food fortification programs to prevent commonly occurring vitamin deficiencies.

Conversely, hyper vitaminosis refers to symptoms caused by vitamin intakes in excess of needs, especially for fat-soluble vitamins that can accumulate in body tissues. The history of the discovery of vitamin deficiencies progressed over centuries from observations that certain conditions for example, scurvy could be prevented or treated with certain foods having high content of a necessary vitamin, to the identification and description of specific molecules essential for life and health. During the 20th century, several scientists were awarded the Nobel Prize in Physiology or Medicine or the Nobel Prize in Chemistry for their roles in the discovery of vitamins.

Vitamin deficiency is the condition of a long-term lack of a vitamin. When caused by not enough vitamin intake it is classified as a primary deficiency, whereas when due to an underlying disorder such as malabsorption it is called a secondary deficiency. An underlying disorder may be metabolic – as in a genetic defect for converting tryptophan to niacin or from lifestyle choices that increase vitamin needs, such as smoking or drinking alcohol. Government guidelines on vitamin deficiencies advise certain intakes for healthy people, with specific values for women, men, babies, the elderly, and during pregnancy or breastfeeding. Many countries have mandated vitamin food fortification programs to prevent commonly occurring vitamin deficiencies.

Conversely, hyper vitaminosis refers to symptoms caused by vitamin intakes in excess of needs, especially for fat-soluble vitamins that can accumulate in body tissues. The history of the discovery of vitamin deficiencies progressed over centuries from observations that certain conditions for example, scurvy could be prevented or treated with certain foods having high content of a necessary vitamin, to the identification and description of specific molecules essential for life and health. During the 20th century, several scientists were awarded the Nobel Prize in Physiology or Medicine or the Nobel Prize in Chemistry for their roles in the discovery of vitamins.

1.1 Problem Statement:

Vitamin deficiencies often go undetected until they manifest as serious health issues. Traditional methods of detection, like blood tests, can be invasive, costly, and inaccessible to some populations. There's a need for a more efficient, non-invasive, and accessible method of early detection to mitigate health risks associated with vitamin deficiencies.

- The challenge at hand is to develop an automated system utilizing deep learning and CNN algorithms that can diagnose potential vitamin deficiencies by analyzing photographs of the eyes, lips, tongue, and nails, eliminating the need for invasive blood tests.

- This research aims to address the lack of a non-invasive, visual-based method for early detection of vitamin deficiencies, necessitating the creation of an accurate and reliable model capable of identifying distinct visual symptoms across different body areas.

1.2 Objectives of the Project:

i. Implement a comprehensive dataset collection and preprocessing strategy for eyes, lips, tongue, and nails images, followed by training and fine-tuning the ANN model to recognize a diverse range of visual symptoms associated with various vitamin deficiencies, leading to an effective and accessible diagnostic tool.

ii. To develop an user-friendly interface, on uploading the images of eyes, lips, tongue and nails detect's the vitamin deficiency of the person.

1.3 Scope:

i.Data Collection and Analysis: Gathering relevant data, possibly from medical records, surveys, and dietary patterns.

ii. Algorithm Development: Developing and training machine learning and deep learning models to identify patterns and indicators of vitamin deficiencies.

iii. Validation and Testing: Rigorously testing the system for accuracy and reliability against traditional detection methods.

CHAPTER 2
LITERATURE SURVEY

[1] Vitamin a deficiency and clinical disease: Vitamin A deficiency has a plethora of clinical manifestations, ranging from xerophthalmia (practically pathognomonic) to disturbances in growth and susceptibility to severe infection (far more protean). Like other classical vitamin deficiency states (scurvy, rickets), some of the signs and symptoms of xerophthalmia were recognized long ago. Reports related to vitamin A and/or manifestations of deficiency might conveniently be divided into "ancient" accounts; eighteenth to nineteenth century clinical descriptions (and their purported etiologic associations); early twentieth century laboratory animal experiments and clinical and epidemiologic observations that identified the existence of this unique nutrient and manifestations of its deficiency; and, most recently, a flowering of carefully conducted clinical studies and field-based randomized trials that documented the full extent and impact of deficiency among the poor of low- and middle-income countries, which in turn changed global health policy.

Summary: Vitamin A deficiency has a plethora of clinical manifestations, ranging from xerophthalmia (practically pathognomonic) to disturbances in growth and susceptibility to severe infection (far more protean). Like other classical vitamin deficiency states (scurvy, rickets), some of the signs and symptoms of xerophthalmia were recognized long ago. Reports related to vitamin A and/or manifestations of deficiency might conveniently be divided into "ancient" accounts; eighteenth to nineteenth century clinical descriptions (and their purported etiologic associations); early twentieth century laboratory animal experiments and clinical and epidemiologic observations that identified the existence of this unique nutrient and manifestations of its deficiency.

[2] Glossitis with linear lesions: The classic oral manifestations of vitamin B(12) deficiency are considered nonspecific. We describe 4 patients with oral linear lesions associated with vitamin B(12) deficiency. Patients were free of neurologic symptoms and anemia at diagnosis. We believe that glossitis with linear lesions is an early clinical sign of vitamin B(12) deficiency. We recommend the determination of vitamin B(12) in such patients, even in the absence of anemia.

Summary: The classic oral manifestations of vitamin B(12) deficiency are considered nonspecific. We describe 4 patients with oral linear lesions associated with vitamin B(12) deficiency.

[3] M. C.Chuang, J. N.Hwang, and C. Rose: Glossodynia, or painful sensation of the tongue, can have a spectrum of etiologies, such as local infection, trauma, nerve damage, glossitis, or the enigmatic neuropathic pain syndrome, burning mouth disorder (BMD; also known as burning mouth syndrome). Careful history-taking, physical examination, and appropriate laboratory screening can differentiate these causes of glossodynia and direct further therapy. A 73-year-old woman presented with several months of glossodynia having previously been diagnosed by her primary care physician with primary BMD. Subsequently, she consulted an otolaryngologist, who pursued further diagnostic evaluation. Examination revealed the presence of a beefy, red, smooth tongue, and further laboratory evaluation yielded a low serum vitamin B(12) level and macrocytosis. Three months of oral vitamin B(12) supplementation led to partial restoration of serum vitamin B(12) levels and a modest improvement in symptoms. Her final diagnoses were atrophic glossitis and glossodynia secondary to vitamin B(12) deficiency, most likely due to pernicious anemia.

Summary: The results of this case have important clinical implications for the diagnostic evaluation and management of patients with glossodynia and apparent BMD. Pathogenic mechanisms of nutrient deficiency in atrophic glossitis are discussed.

CHAPTER 3

METHODOLOGY

As you can see, each image are presented in the matrix formats, which are made up of rows and columns. The pixel is an image's fundamental building block. A group of pixels make up an image. These are all little squares. We may build the entire image by arranging them side by side. The smallest amount of information that can be present in an image is a single pixel. Every image has pixels with values ranging from 0 to 255.

Each pixel is composed of Three values are R, G, and B, which are the basic colours red, green, and blue. The combination of these three basic colours will create all these colours here in the image so we conclude that a single pixel has three channels, one channel for each one of the basic colours.

3.1 Machine Learning

Machine Learning is undeniably one of the most influential and powerful technologies in today‘s world. Machine learning is a tool for turning information into knowledge. In the past 50 years, there has been an explosion of data. This mass of data is useless; we analyse it and find the patterns hidden within. Machine learning techniques are used to automatically find the valuable underlying patterns within complex data that we would otherwise struggle to discover.

The hidden patterns and knowledge about a problem can be used to predict future events and perform all kinds of complex decision making. To learn the rules governing a phenomenon, machines have to go through a learning process**,** trying different rules and learning from how well they perform. Hence, why it‘s known as Machine Learning.

Basic Terminology:

- Dataset: A set of data examples, which contain features important to solving the problem.
- Features: Important pieces of data that help us understand a problem. These are fed into a Machine Learning algorithm to help it learn.

- Model: The representation (internal model) of a phenomenon that a Machine Learning algorithm has learnt. It learns this from the data it is shown during training. The model is the output you get after training an algorithm. For example, a decision tree algorithm would be trained and produce a decision tree model.

Types of Machine Learning:

There are multiple forms of Machine Learning; supervised, unsupervised, semi supervised and reinforcement learning. Each form of Machine Learning has differing approaches, but they all follow the same underlying process and theory.

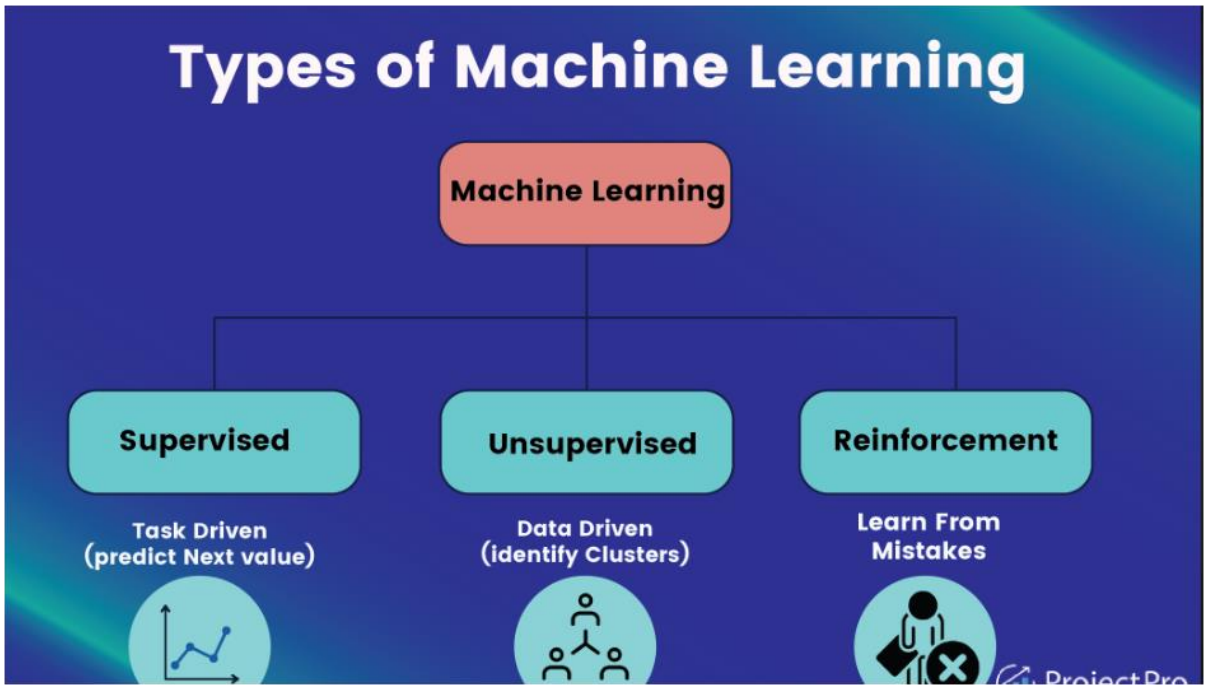

Fig 3.1: Types of Machine Learning

Supervised Learning: It is the most popular paradigm for machine learning. Given data in the form of examples with labels, we can feed a learning algorithm these example-label pairs one by one, allowing the algorithm to predict the label for each example, and giving it feedback as to whether it predicted the right answer or not. Over time, the algorithm will learn to approximate the exact nature of the relationship between examples and their labels. When fully-trained, the supervised learning algorithm will be able to observe a new, never before-seen example and predict a good label for it.

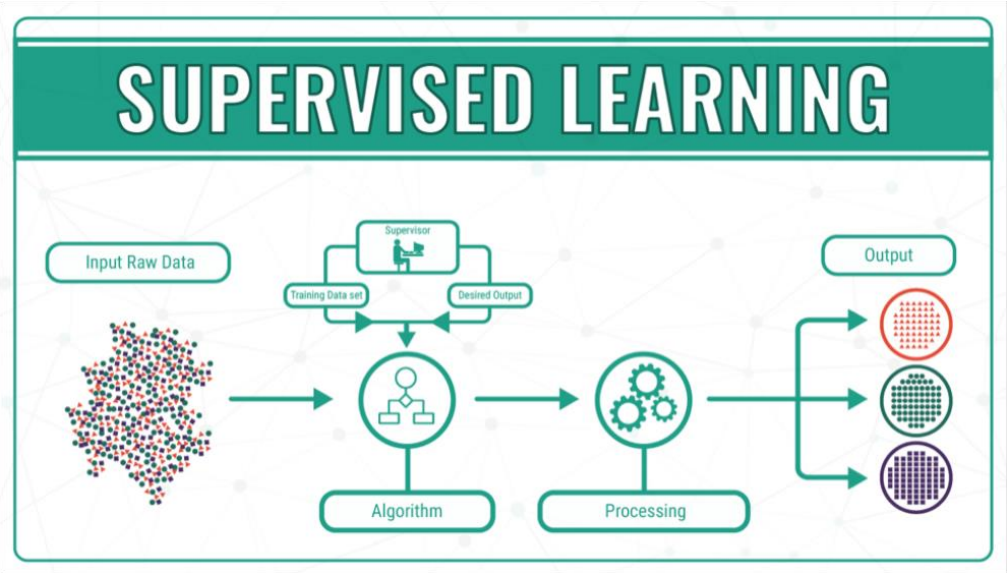

Fig 3.2: Process of Supervised Learning

Unsupervised learning: It is very much the opposite of supervised learning. It features no labels. Instead, the algorithm would be fed a lot of data and given the tools to understand the properties of the data. From there, it can learn to group, cluster, and organize the data in a way such that a human can come in and make sense of the newly organized data. Because unsupervised learning is based upon the data and its properties, we can say that unsupervised learning is data- driven. The outcomes from an unsupervised learning task are controlled by the data and the way it‘s formatted.

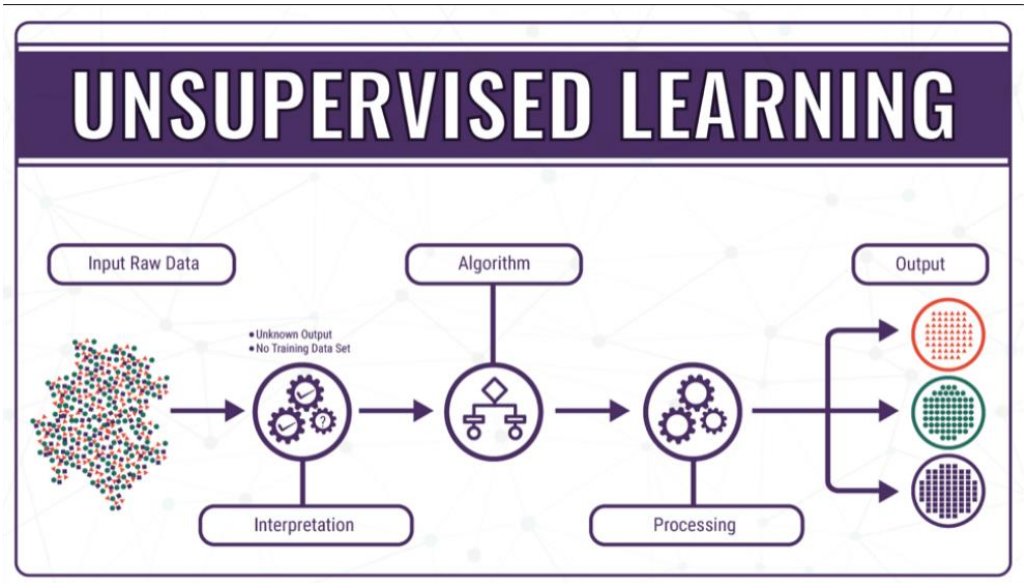

Fig 3.3: Process of Unsupervised Learning

Reinforcement learning: It is fairly different when compared to supervised and unsupervised learning. Reinforcement learning is very behaviour driven. It has influences from the fields of neuroscience and psychology. For any reinforcement learning problem, we need an agent and an environment as well as a way to connect the two through a feedback loop. To connect the agent to the environment, we give it a set of actions that it can take that affect the environment. To connect the environment to the agent, we have it continually issue two signals to the agent: an updated state and a reward (our reinforcement signal for behaviour).

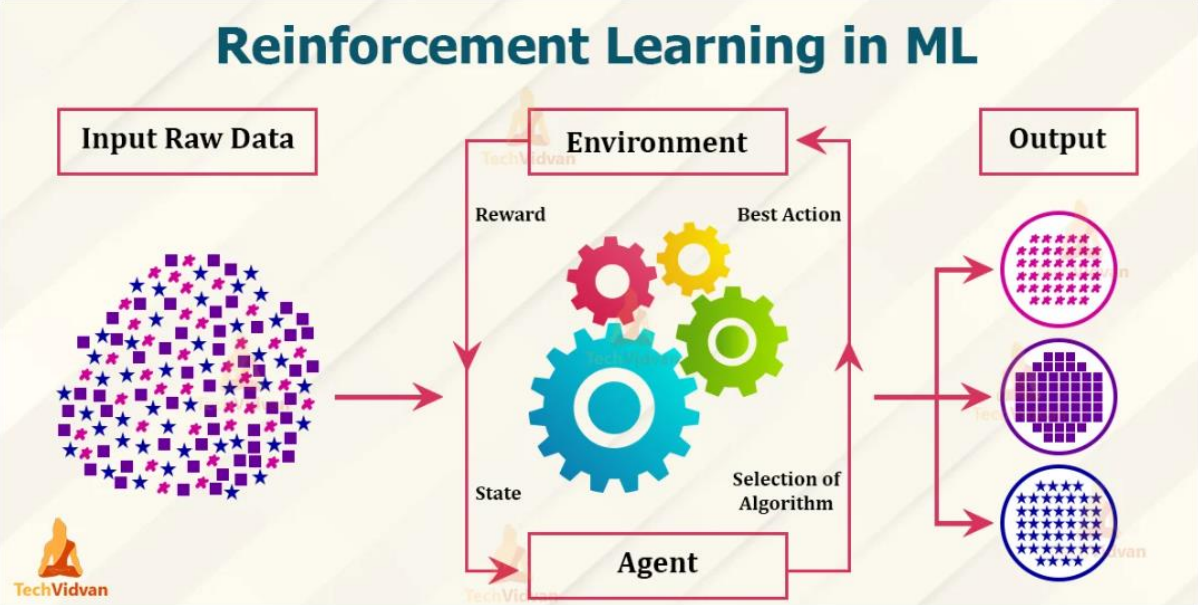

Fig 3.4: Reinforcement Learning

3.2. Algorithm Used

Convolutional Neural Network

Step1: convolutional operation

The first building block in our plan of attack is convolution operation. In this step, we will touch on feature detectors, which basically serve as the neural network's filters. We will also discuss feature maps, learning the parameters of such maps, how patterns are detected, the layers of detection, and how the findings are mapped out.

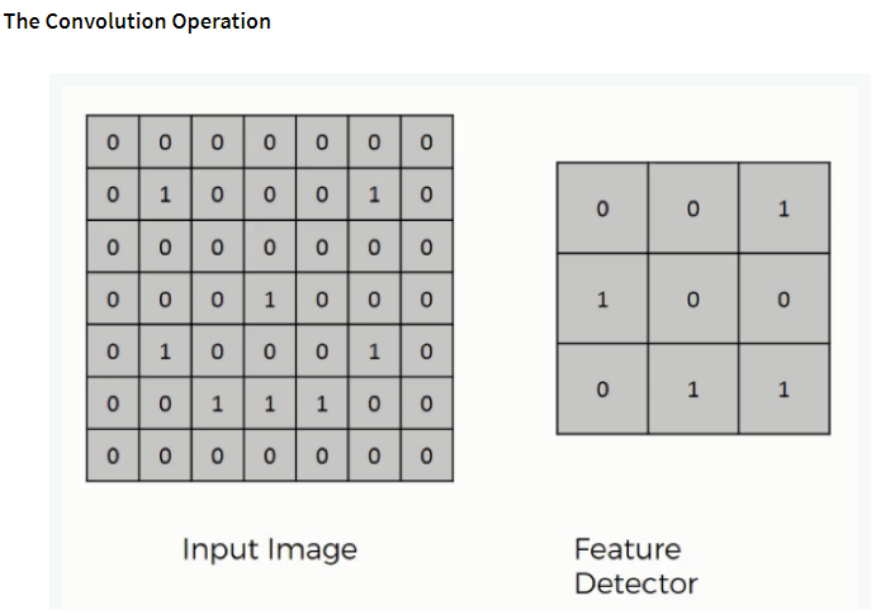

Fig 3.5 Convolution Operation

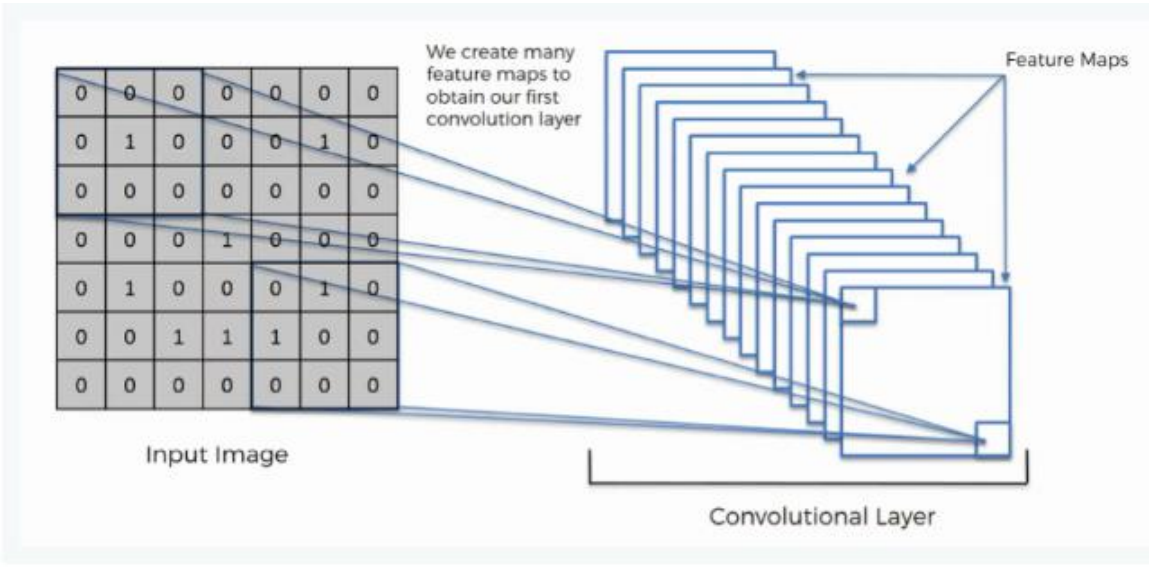

Fig 3.6 Feature Maps

Step (1b): ReLU Layer

The second part of this step will involve the Rectified Linear Unit or Relook. We will cover Relook layers and explore how linearity functions in the context of Convolutional Neural Networks.

Not necessary for understanding CNN's, but there's no harm in a quick lesson to improve your skills.

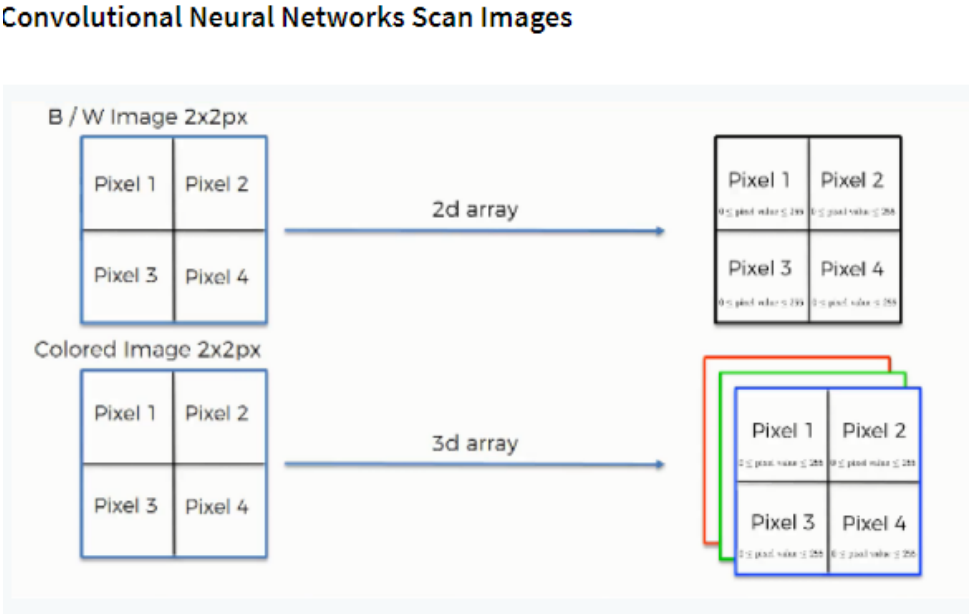

Fig 3.7 Image Scan

Step 2: Pooling Layer

In this part, we'll cover pooling and will get to understand exactly how it generally works. Our nexus here, however, will be a specific type of pooling; max pooling. We'll cover various approaches, though, including mean (or sum) pooling. This part will end with a demonstration made using a visual interactive tool that will definitely sort the whole concept out for you.

Step 3: Flattening

This will be a brief breakdown of the flattening process and how we move from pooled to flattened layers when working with Convolutional Neural Networks.

Step 4: Full Connection

In this part, everything that we covered throughout the section will be merged together. By learning this, you'll get to envision a fuller picture of how Convolutional Neural Networks operate and how the "neurons" that are finally produced learn the classification of images.

Summary

In the end, we'll wrap everything up and give a quick recap of the concept covered in the section. If you feel like it will do you any benefit (and it probably will), you should check out the extra tutorial in which Softmax and Cross-Entropy are covered. It's not mandatory for the course, but you will likely come across these concepts when working with Convolutional Neural Networks and it will do you a lot of good to be familiar with them.

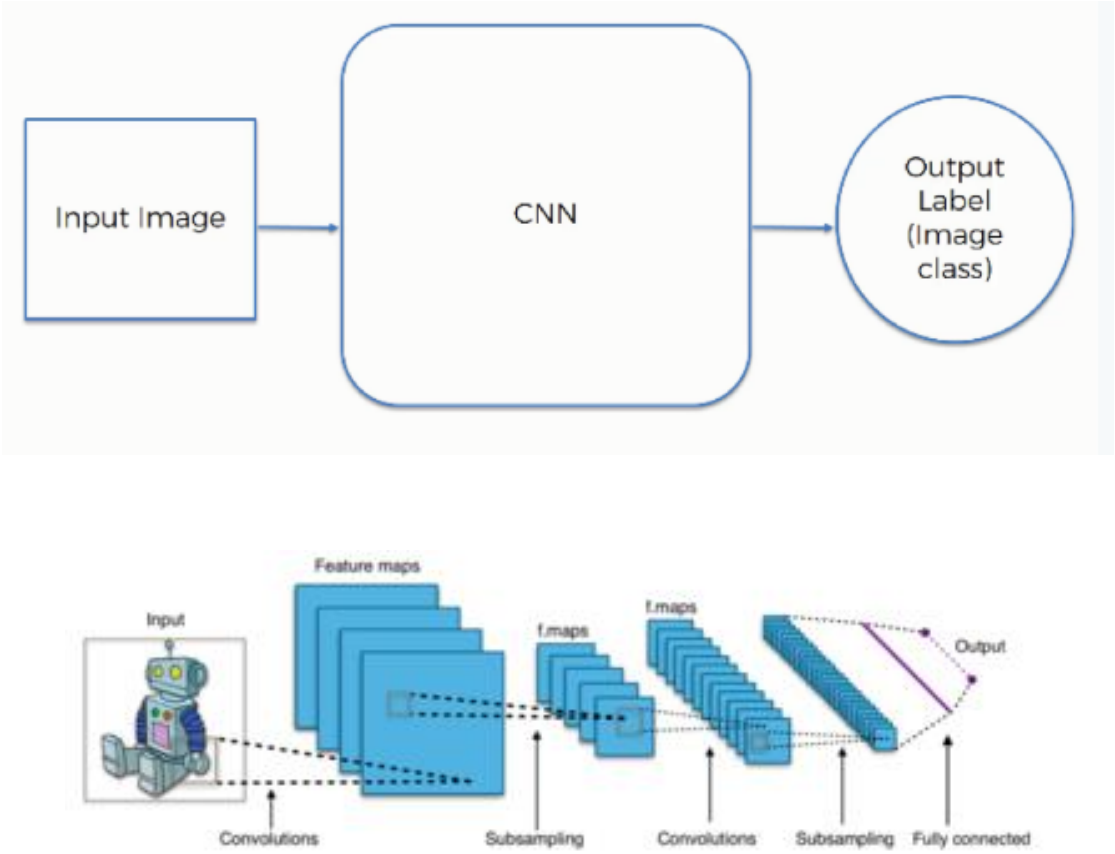

Fig 3.8 CNN Architecture

CHAPTER 4

SYSTEM REQUIREMENT SPECIFICATION

4.1 Functional Requirements

Functional requirements specify what the system should do. They include:

Data Collection and Processing: Ability to collect various types of data (dietary patterns, physical symptoms, patient history, etc.).Efficient preprocessing of data to format it suitably for analysis.

Algorithm Implementation: Developing and integrating machine learning and deep learning algorithms capable of identifying signs of vitamin deficiencies. Implementing algorithms that can learn from new data to improve accuracy over time.

User Interface: A user-friendly interface for both healthcare professionals and patients. Features for users to input data and view results.

Diagnostic Reporting: Generating comprehensive reports indicating the likelihood of vitamin deficiencies. Providing recommendations or alerts based on the analysis.

Data Management: Secure storage of all user data and analysis results. Functionality for data retrieval, update, and deletion as per user request or legal compliance.

Integration Capabilities: Compatibility with existing healthcare systems and databases.

APIs or other means to integrate with electronic health records (EHR) systems.

4.2 Non-Functional Requirements

Non-functional requirements define how the system operates and include:

Performance: High accuracy and reliability in detecting vitamin deficiencies. Fast processing and response time.

Scalability: Capability to handle an increasing amount of data and users without performance degradation.

Usability: Ease of use for individuals with varying levels of tech proficiency. Accessibility features for users with disabilities.

Security and Privacy: Robust security measures to protect sensitive health data. Compliance with data protection regulations (like GDPR, HIPAA, etc.).

Maintainability and Support: Ease of updating and maintaining the system. Availability of technical support and user documentation.

Compatibility: Compatibility with various devices and operating systems. Interoperability with different data formats and healthcare IT systems.

4.3 Python Libraries

SOFTWARE INSTALLATION FOR MACHINE LEARNING PROJECTS:

Installing Python:

1. To download and install Python visit the official website of Python https://www.python.org/downloads/ and choose your version.

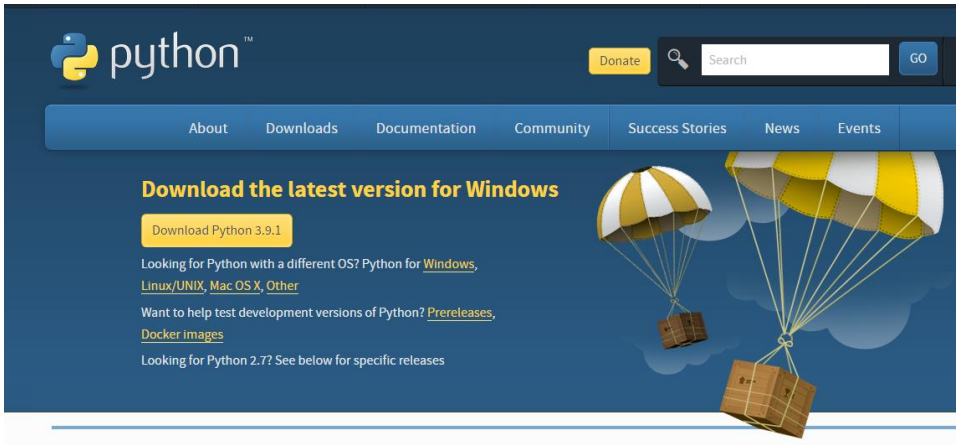

Fig 4.1 Python Installation

2. Once the download is complete, run the exe for install Python. Now click on Install Now.

3. You can see Python installing at this point.

4. When it finishes, you can see a screen that says the Setup was successful. Now click on "Close".

Installing PyCharm:

1. To download PyCharm visit the website https://www.jetbrains.com/pycharm/download/ and click the "DOWNLOAD" link under the Community Section.
2. Once the download is complete, run the exe for install PyCharm. The setup wizard should have started. Click “Next”.
3. On the next screen, Change the installation path if required. Click “Next”.
4. On the next screen, you can create a desktop shortcut if you want and click on “Next”.
5. Choose the start menu folder. Keep selected Jet Brains and click on “Install”.
6. Wait for the installation to finish.
7. Once installation finished, you should receive a message screen that PyCharm is installed. If you want to go ahead and run it, click the “Run PyCharm Community Edition” box first and click “Finish”.
8. After you click on "Finish," the Following screen will appear.
9. You need to install some packages to execute your project in a proper way.
10. Open the command prompt/ anaconda prompt or terminal as administrator.
11. The prompt will get open, with specified path, type “pip install package name” which you want to install (like NumPy, pandas, sea born, scikit-learn, Matplotlib, Pyplot)

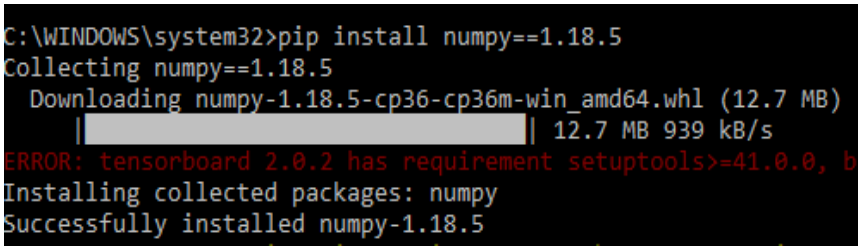

Fig 4.2 Pip install NumPy

INTRODUCTION TO PYTHON

What Is a Script?

Up to this point, I have concentrated on the interactive programming capability of Python. This is a very useful capability that allows you to type in a program and to have it executed immediately in an interactive mode

Scripts are reusable: Basically, a script is a text file containing the statements that comprise a Python program. Once you have created the script, you can execute it over and over without having to retype it each time.

Scripts are editable:

Perhaps, more importantly, you can make different versions of the script by modifying the statements from one file to the next using a text editor. Then you can execute each of the individual versions. In this way, it is easy to create different programs with a minimum amount of typing.

You will need a text editor:

Just about any text editor will suffice for creating Python script files.

You can use *Microsoft Notepad*, *Microsoft WordPad*, *Microsoft Word*, or just about any word processor if you want to.

Difference between a script and a program:

Scripts are distinct from the core code of the application, which is usually written in a different language, and are often created or at least modified by the end-user. Scripts are often interpreted from source code or byte code, whereas the applications they control are traditionally compiled to native machine code.The program has an executable form that the computer can use directly to execute the instructions. The same program in its human-readable source code form, from which executable programs are derived (e.g., compiled)

Python:

What is Python? Chances you are asking yourself this. You may have found this book because you want to learn to program but don't know anything about programming languages. Or you may have heard of programming languages like C, C++, C#, or Java and want to know what Python is and how it compares to "big name" languages. Hopefully I can explain it for you. If you're not interested in the how's and whys of Python, feel free to skip to the next chapter. In this chapter I will try to explain to the reader why I think Python is one of the best languages available and why it's a great one to start programming with.

• Open-source general-purpose language.

• Object Oriented, Procedural, Functional

• Easy to interface with C/ObjC/Java/Fortran

• Easy-is to interface with C++ (via SWIG)

• Great interactive environment

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.

Python is Interactive − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

Python is Object-Oriented − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.

Python is a Beginner's Language − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

History of Python:

Python was developed by Guido van Possum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands. Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, Smalltalk, and UNIX shell and other scripting languages. Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL). Python is now maintained by a core development team at the institute, although Guido van Possum still holds a vital role in directing its progress.

Python Features:

Python's features include −

1. Easy-to-learn − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
2. Easy-to-read − Python code is more clearly defined and visible to the eyes.
3. Easy-to-maintain − Python's source code is fairly easy-to-maintained.
4. A broad standard library − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
5. Interactive Mode − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
6. Portable − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
7. Extendable − you can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
8. Databases − Python provides interfaces to all major commercial databases.
9. GUI Programming − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
10. Scalable − Python provides a better structure and support for large programs than shell scripting.
11. Apart from the above-mentioned features, Python has a big list of good features, few are listed below −
12. It supports functional and structured programming methods as well as OOP.
13. It can be used as a scripting language or can be compiled to byte-code for building large applications.
14. It provides very high-level dynamic data types and supports dynamic type checking.
15. IT supports automatic garbage collection.
16. It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

Dynamic vs. Static:

Types Python is a dynamic-typed language. Many other languages are static typed, such as C/C++ and Java. A static typed language requires the programmer to explicitly tell the computer what type of "thing" each data value is.

For example, in C if you had a variable that was to contain the price of something, you would have to declare the variable as a "float" type.

This tells the compiler that the only data that can be used for that variable must be a floating-point number, i.e. a number with a decimal point.

If any other data value was assigned to that variable, the compiler would give an error when trying to compile the program.

Python, however, doesn't require this. You simply give your variables names and assign values to them. The interpreter takes care of keeping track of what kinds of objects your program is using. This also means that you can change the size of the values as you develop the program. Say you have another decimal number (a.k.a. a floating-point number) you need in your program.

With a static typed language, you have to decide the memory size the variable can take when you first initialize that variable. A double is a floating-point value that can handle a much larger number than a normal float (the actual memory sizes depend on the operating environment).

If you declare a variable to be a float but later on assign a value that is too big to it, your program will fail; you will have to go back and change that variable to be a double.

With Python, it doesn't matter. You simply give it whatever number you want and Python will take care of manipulating it as needed. It even works for derived values.

For example, say you are dividing two numbers. One is a floating-point number and one is an integer. Python realizes that it's more accurate to keep track of decimals so it automatically calculates the result as a floating-point number

Variables:

Variables are nothing but reserved memory locations to store values. This means that when you create a variable you reserve some space in memory.

Based on the data type of a variable, the interpreter allocates memory and decides what can be stored in the reserved memory. Therefore, by assigning different data types to variables, you can store integers, decimals or characters in these variables.

Standard Data Types:

The data stored in memory can be of many types. For example, a person's age is stored as a numeric value and his or her address is stored as alphanumeric characters. Python has various standard data types that are used to define the operations possible on them and the storage method for each of them.

Python has five standard data types −

1. Numbers
2. String
3. List
4. Tuple
5. Dictionary

Python Numbers:

Number data types store numeric values. Number objects are created when you assign a value to them

Python Strings:

Strings in Python are identified as a contiguous set of characters represented in the quotation marks. Python allows for either pairs of single or double quotes. Subsets of strings can be taken using the slice operator ([ ] and [:]) with indexes starting at 0 in the beginning of the string and working their way from -1 at the end.

Python Lists:

Lists are the most versatile of Python's compound data types. A list contains items separated by commas and enclosed within square brackets ([]). To some extent, lists are similar to arrays in C. One difference between them is that all the items belonging to a list can be of different data type.

The values stored in a list can be accessed using the slice operator ([ ] and [:]) with indexes starting at 0 in the beginning of the list and working their way to end -1. The plus (+) sign is the list concatenation operator, and the asterisk (*) is the repetition operator.

Python Tuples:

A tuple is another sequence data type that is similar to the list. A tuple consists of a number of values separated by commas. Unlike lists, however, tuples are enclosed within parentheses.

The main differences between lists and tuples are: Lists are enclosed in brackets ([ ]) and their elements and size can be changed, while tuples are enclosed in parentheses (( )) and cannot be updated. Tuples can be thought of as read-only lists.

Python Dictionary:

Python's dictionaries are kind of hash table type. They work like associative arrays or hashes found in Perl and consist of key-value pairs. A dictionary key can be almost any Python type, but are usually numbers or strings. Values, on the other hand, can be any arbitrary Python object.

Dictionaries are enclosed by curly braces ({ }) and values can be assigned and accessed using square braces ([]).

Different modes in python:

Python has two basic modes: normal and interactive.

The normal mode is the mode where the scripted and finished .pie files are run in the Python interpreter.

Interactive mode is a command line shell which gives immediate feedback for each statement, while running previously fed statements in active memory. As new lines are fed into the interpreter, the fed program is evaluated both in part and in whole

20 Python libraries:

1. Requests. The most famous http library written by Kenneth remits. It’s a must have for every python developer.

2. Scrappy. If you are involved in web scraping then this is a must have library for you. After using this library you won’t use any other.

3. Python. A guy toolkit for python. I have primarily used it in place of tinder. You will really love it.

4. Pillow. A friendly fork of PIL (Python Imaging Library). It is more user friendly than PIL and is a must have for anyone who works with images.

5. SQL Alchemy. A database library. Many love it and many hate it. The choice is yours.

6. Beautiful Soup. I know it's slow but this xml and html parsing library is very useful for beginners.

7. Twisted. The most important tool for any network application developer. It has a very beautiful ape and is used by a lot of famous python developers.

8. Numbly. How can we leave this very important library? It provides some advance math functionalities to python.

9. Skippy. When we talk about numbly then we have to talk about spicy. It is a library of algorithms and mathematical tools for python and has caused many scientists to switch from ruby to python.

10. Matplotlib. A numerical plotting library. It is very useful for any data scientist or any data analyzer.

11. Pygmy. Which developer does not like to play games and develop them? This library will help you achieve your goal of 2d game development.

12. Piglet. A 3d animation and game creation engine. This is the engine in which the famous python port of mine craft was made

13. Pit. A GUI toolkit for python. It is my second choice after python for developing GUI's for my python scripts.

14. Pit. Another python GUI library. It is the same library in which the famous Bit torrent client is created.

15. Scaly. A packet sniffer and analyser for python made in python.

16. Pywin32. A python library which provides some useful methods and classes for interacting with windows.

17. Notch. Natural Language Toolkit – I realize most people won't be using this one, but it's generic enough. It is a very useful library if you want to manipulate strings. But its capacity is beyond that. Do check it out.

18. Nose. A testing framework for python. It is used by millions of python developers. It is a must have if you do test driven development.

19. Simply. Simply can do algebraic evaluation, differentiation, expansion, complex numbers, etc. It is contained in a pure Python distribution.

20. Python can't stress enough how useful this tool is. It is a python prompt on steroids. It has completion, history, shell capabilities, and a lot more. Make sure that you take a look at it.

Numpy:

Numpy's main object is the homogeneous multidimensional array. It is a table of elements (usually numbers), all of the same type, indexed by a tuple of positive integers. In numbly dimensions are called *axes*. The number of axes is *rank*.

• Offers Matlab-ish capabilities within Python

• Fast array operations

• 2D arrays, multi-D arrays, linear algebra etc.

Matplotlib:

• High quality plotting library.

Python class and objects:

These are the building blocks of OOP. Class creates a new object. This object can be anything, whether an abstract data concept or a model of a physical object, e.g. a chair. Each class has individual characteristics unique to that class, including variables and methods. Classes are very powerful and currently "the big thing" in most programming languages. Hence, there are several chapters dedicated to OOP later in the book. The class is the most basic component of object-oriented programming. Previously, you learned how to use functions to make your program do something. Now will move into the big, scary world of Object-Oriented Programming (OOP). To be honest, it took me several months to get a handle on objects. When I first learned C and C++, I did great; functions just made sense for me. Having messed around with BASIC in the early '90s, I realized functions were just like subroutines so there wasn't much new to learn. However, when my C++ course started talking about objects, classes, and all the new features of OOP, my grades definitely suffered. Once you learn OOP, you'll realize that it's actually a pretty powerful tool. Plus many Python libraries and APIs use classes, so you should at least be able to understand what the code is doing. One thing to note about Python and OOP: it's not mandatory to use objects in your code in a way that works best; maybe you don't need to have a full-blown class with initialization code and methods to just return a calculation.

Python, you can get as technical as you want. As you've already seen, Python can do just fine with functions. Unlike languages such as Java, you aren't tied down to a single way of doing things; you can mix functions and classes as necessary in the same program. This lets

you build the code Objects are an encapsulation of variables and functions into a single entity. Objects get their variables and functions from classes. Classes are essentially a template to create your objects.

Here's a brief list of Python OOP ideas:

The class statement creates a class object and gives it a name. This creates a new namespace. Assignments within the class create class attributes. These attributes are accessed by qualifying the name using dot syntax: Class Name .Attribute. Class attributes export the state of an object and its associated behavior. These attributes are shared by all instances of a class. Calling a class (just like a function) creates a new instance of the class. This is where the multiple copies part comes in. Each instance gets ("inherits") the default class attributes and gets its own namespace. This prevents instance objects from overlapping and confusing the program. Using the term self identifies a particular instance, allowing for per-instance attributes. This allows items such as variables to be associated with a particular instance.

Inheritance:

First off, classes allow you to modify a program without really making changes to it. To elaborate, by sub classing a class, you can change the behaviour of the program by simply adding new components to it rather than rewriting the existing components. As we've seen, an instance of a class inherits the attributes of that class. However, classes can also inherit attributes from other classes. Hence, a subclass inherits from a superclass allowing you to make a generic superclass that is specialized via subclasses. The subclasses can override the logic in a superclass, allowing you to change the behavior of your classes without changing the superclass at all.

Operator Overloads:

Operator overloading simply means that objects that you create from classes can respond to actions (operations) that are already defined within Python, such as addition, slicing, printing, etc. Even though these actions can be implemented via class methods, using overloading ties the behaviour closer to Python's object model and the object interfaces are more consistent to Python's built-in objects, hence overloading is easier to learn and use. User-made classes can override nearly all of Python's built-in operation methods

Exceptions :

I've talked about exceptions before but now I will talk about them in depth. Essentially, exceptions are events that modify program's flow, either intentionally or due to errors. They are special events that can occur due to an error, e.g. trying to open a file that doesn't exist, or when the program reaches a marker, such as the completion of a loop. Exceptions, by definition, don't occur very often; hence, they are the "exception to the rule" and a special class has been created for them. Exceptions are everywhere in Python. Virtually every module in the standard Python library uses them, and Python itself will raise them in a lot of different circumstances.

Here are just a few examples:

• Accessing a non−existent dictionary key will raise a Key Error exception.

• Searching a list for a non−existent value will raise a Value Error exception

. • Calling a non−existent method will raise an Attribute Error exception.

• Referencing a non−existent variable will raise a Name Error exception.

• Mixing data types without coercion will raise a Type Error exception.

One use of exceptions is to catch a fault and allow the program to continue working; we have seen this before when we talked about files. This is the most common way to use exceptions. When programming with the Python command line interpreter, you don't need to worry about catching exceptions.

Your program is usually short enough to not be hurt too much if an exception occurs. Plus, having the exception occur at the command line is a quick and easy way to tell if your code logic has a problem. However, if the same error occurred in your real program, it will fail and stop working. Exceptions can be created manually in the code by raising an exception. It operates exactly as a system-caused exceptions, except that the programmer is doing it on purpose. This can be for a number of reasons. One of the benefits of using exceptions is that, by their nature, they don't put any overhead on the code processing. Because exceptions aren't supposed to happen very often, they aren't processed until they occur. Exceptions can be thought of as a special form of the if/elf statements. You can realistically do the same thing with if blocks as you can with exceptions. However, as already mentioned, exceptions aren't processed until they occur; if blocks are processed all the time. Proper use of exceptions can

help the performance of your program. The more infrequent the error might occur, the better off you are to use exceptions; using if blocks requires Python to always test extra conditions before continuing. Exceptions also make code management easier: if your programming logic is mixed in with error-handling if statements, it can be difficult to read, modify, and debug your program.

User-Defined Exceptions:

I won't spend too much time talking about this, but Python does allow for a programmer to create his own exceptions. You probably won't have to do this very often but it's nice to have the option when necessary. However, before making your own exceptions, make sure there isn't one of the built-in exceptions that will work for you. They have been "tested by fire" over the years and not only work effectively, they have been optimized for performance and are bug-free.

Making your own exceptions involves object-oriented programming, which will be covered in the next chapter. To make a custom exception, the programmer determines which base exception to use as the class to inherit from, e.g. making an exception for negative numbers or one for imaginary numbers would probably fall under the Arithmetic Error exception class. To make a custom exception, simply inherit the base exception and define what it will do.

Python modules:

Python allows us to store our code in files (also called modules). This is very useful for more serious programming, where we do not want to retype a long function definition from the very beginning just to change one mistake. In doing this, we are essentially defining our own modules, just like the modules defined already in the Python library. To support this, Python has a way to put definitions in a file and use them in a script or in an interactive instance of the interpreter. Such a file is called a *module*; definitions from a module can be *imported* into other modules or into the *main* module.

Testing code:

As indicated above, code is usually developed in a file using an editor.To test the code, import it into a Python session and try to run it.Usually there is an error, so you go back to the file, make a correction, and test again. This process is repeated until you are satisfied that the code works. This entire process is known as the development cycle. There are two types of

errors that you will encounter. Syntax errors occur when the form of some command is invalid. This happens when you make typing errors such as misspellings, or call something by the wrong name, and for many other reasons. Python will always give an error message for a syntax error.

Functions in Python:

It is possible, and very useful, to define our own functions in Python. Generally speaking, if you need to do a calculation only once, then use the interpreter. But when you or others have need to perform a certain type of calculation many times, then define a function.

You use functions in programming to bundle a set of instructions that you want to use repeatedly or that, because of their complexity, are better self-contained in a sub-program and called when needed. That means that a function is a piece of code written to carry out a specified task. To carry out that specific task, the function might or might not need multiple inputs. When the task is carved out, the function can or cannot return one or more values.

There are three types of functions in python:

Help (), min (), print ().

Namespaces in Python are implemented as Python dictionaries, this means it is a mapping from names (keys) to objects (values). The user doesn't have to know this to write a Python program and when using namespaces.

- Some namespaces in Python:
- global names of a module
- local names in a function or method invocation
- built-in names: this namespace contains built-in functions (e.g. abs(), camp(), ...) and built-in exception names

Garbage Collection:

Garbage Collector exposes the underlying memory management mechanism of Python, the automatic garbage collector. The module includes functions for controlling how the collector operates and to examine the objects known to the system, either pending collection or stuck in reference cycles and unable to be freed.

Python XML Parser:

XML is a portable, open source language that allows programmers to develop applications that can be read by other applications, regardless of operating system and/or developmental language.

What is XML? The Extensible Markup Language XML is a markup language much like HTML or SGML. This is recommended by the World Wide Web Consortium and available as an open standard. XML is extremely useful for keeping track of small to medium amounts of data without requiring a SQL-based backbone. XML Parser Architectures and APIs the Python standard library provides a minimal but useful set of interfaces to work with XML. The two most basic and broadly used APIs to XML data are the SAX and DOM interfaces. Simple API for XML SAX: Here, you register call-backs for events of interest and then let the parser proceed through the document.

This is useful when your documents are large or you have memory limitations, it parses the file as it reads it from disk and the entire file is never stored in memory.

Document Object Model DOM API : This is a World Wide Web Consortium recommendation wherein the entire file is read into memory and stored in a hierarchical tree − based form to represent all the features of an XML document. SAX obviously cannot process information as fast as DOM can when working with large files. On the other hand, using DOM exclusively can really kill your resources, especially if used on a lot of small files. SAX is read-only, while DOM allows changes to the XML file. Since these two different APIs literally complement each other, there is no reason why you cannot use them both for large projects.

Python Web Frameworks:

A web framework is a code library that makes a developer's life easier when building reliable, scalable and maintainable web applications.

Why are web frameworks useful?

Web frameworks encapsulate what developers have learned over the past twenty years while programming sites and applications for the web. Frameworks make it easier to reuse code for common HTTP operations and to structure projects so other developers with knowledge of the framework can quickly build and maintain the application.

Common web framework functionality are Frameworks provide functionality in their code or through extensions to perform common operations required to run web applications. These common operations include:

I. URL routing
II. HTML, XML, JSON, and other output format tinplating
III. Database manipulation
IV. Security against Cross-site request forgery (CSRF) and other attacks
V. Session storage and retrieval
VI. Not all web frameworks include code for all of the above functionality. Frameworks fall on the spectrum from executing a single use case to providing every known web framework feature to every developer. Some frameworks take the "batteries-included" approach where everything possible comes bundled with the framework while others have a minimal core package that is amenable to extensions provided by other packages.

Comparing web frameworks:

There is also a repository called compare-python-web-frameworks where the same web application is being coded with varying Python web frameworks, tinplating engines and object. Web framework resources.

- When you are learning how to use one or more web frameworks it's helpful to have an idea of what the code under the covers is doing.
- Frameworks is a really well done short video that explains how to choose between web frameworks. The author has some particular opinions about what should be in a framework. For the most part I agree although I've found sessions and database ORMs to be a helpful part of a framework when done well.
- What is a web framework? Is an in-depth explanation of what web frameworks are and their relation to web servers?
- Jingo vs. Flash vs. Pyramid: Choosing a Python web framework contains background information and code comparisons for similar web applications built in these three big Python frameworks.
- This fascinating blog post takes a look at the code complexity of several Python web frameworks by providing visualizations based on their code bases.
- What web frameworks do you use and why are they awesome? Is a language agnostic Reedit discussion on web frameworks? It's interesting to see what programmers in other

languages like and dislike about their suite of web frameworks compared to the main Python frameworks.

- This user-voted question & answer site asked "What are the best general purpose Python web frameworks usable in production?" The votes aren't as important as the list of the many frameworks that are available to Python developers.

Web frameworks learning checklist

1. Choose a major Python web framework (Jingo or Flask are recommended) and stick with it. When you're just starting it's best to learn one framework first instead of bouncing around trying to understand every framework.
2. Work through a detailed tutorial found within the resources links on the framework's page.
3. Study open source examples built with your framework of choice so you can take parts of those projects and reuse the code in your application.
4. Build the first simple iteration of your web application then go to the deployment section to make it accessible on the web.

4.4 Software Requirements:

- Operating System : Windows 7/8/10
- Server side Script : HTML, CSS, Bootstrap & JS
- Programming Language : Python
- Libraries : Flask, Pandas, Mysql.connector, Os, Smtplib, Numpy
- IDE/Workbench : PyCharm
- Technology : Python 3.6+
- Server Deployment : Xampp Server
- Database : MySQL

4.5 Hardware Requirements

- Processor - I3/Intel Processor
- Hard Disk - 160GB
- Key Board - Standard Windows Keyboard
- Mouse - Two or Three Button Mouse
- Monitor - SVGA
- RAM -8GB

CHAPTER 5

SYSTEM ANALYSIS AND DESIGN

5.1 UML DIAGRAMS:

5.1.1 Use Case Diagram:

- A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis.
- Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases.
- The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

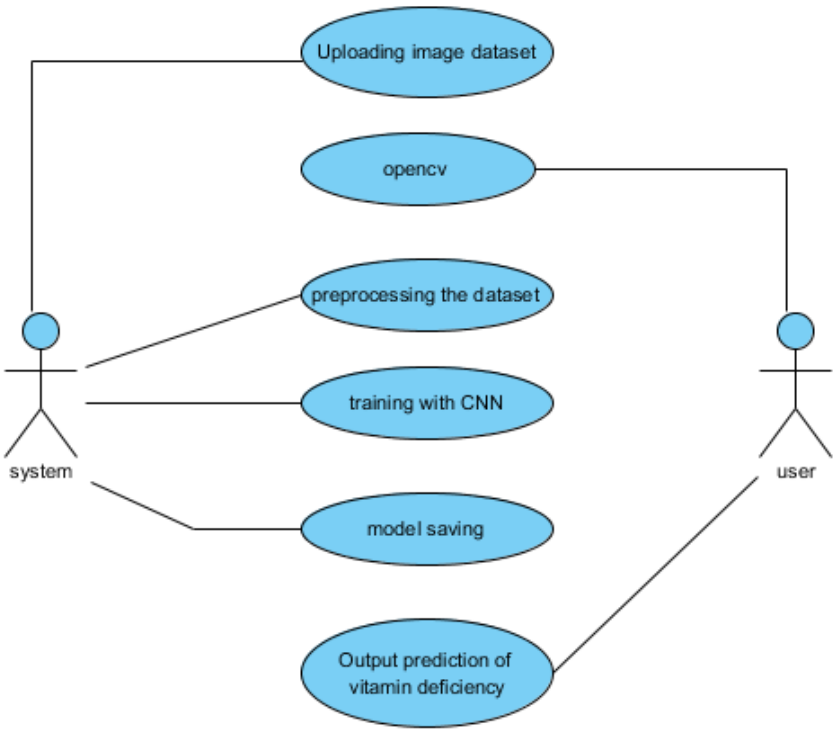

Fig 5.1: Usecase Diagram

5.1.2 Class Diagram:

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information

Fig 5.2: Class Diagram

5.1.3 Sequence Diagram:

- A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order.
- It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

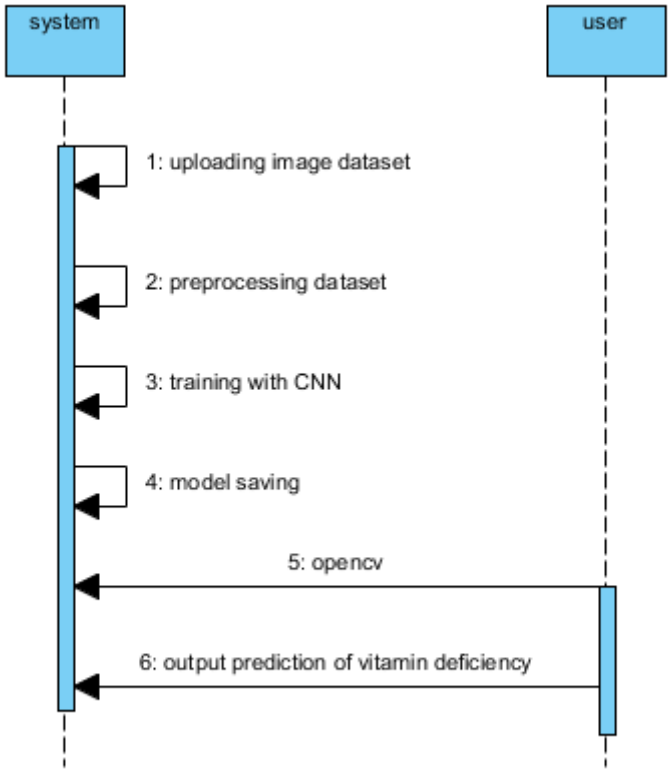

Fig 5.3: Sequence Diagram

5.1.4 Collaboration Diagram:

In collaboration diagram the method call sequence is indicated by some numbering technique as shown below. The number indicates how the methods are called one after another. We have taken the same order management system to describe the collaboration diagram. The method calls are similar to that of a sequence diagram. But the difference is that the sequence diagram does not describe the object organization whereas the collaboration diagram shows the object organization.

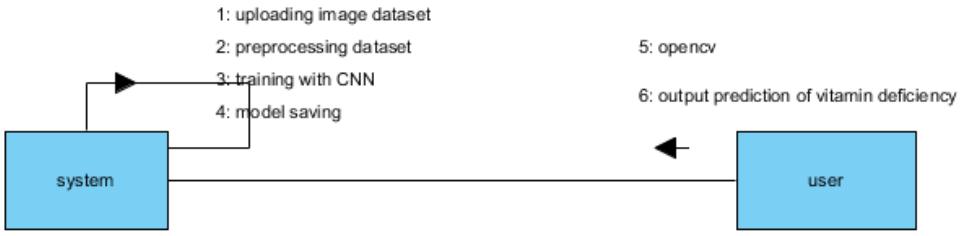

Fig 5.4: Collaboration Diagram

5.1.5 Deployment Diagram

Deployment diagram represents the deployment view of a system. It is related to the component diagram. Because the components are deployed using the deployment diagrams. A deployment diagram consists of nodes. Nodes are nothing but physical hardware's used to deploy the application.

5.1.6 Activity Diagram:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

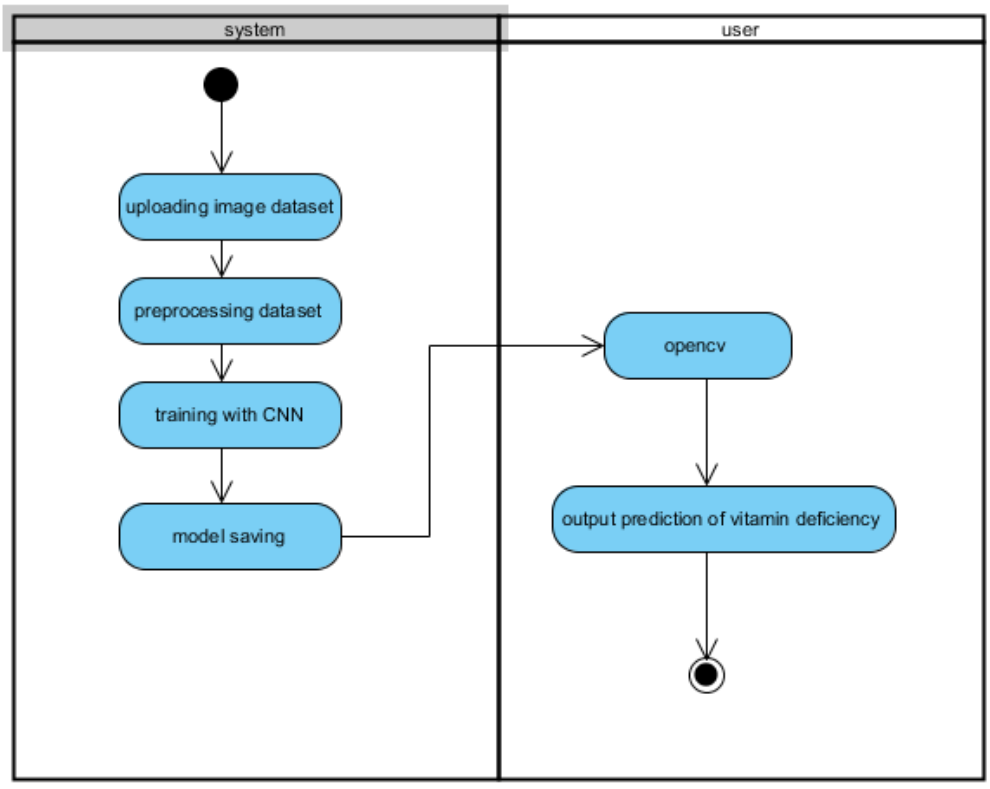

Fig 5.5: Activity Diagram

5.1.7 Component Diagram:

A component diagram, also known as a UML component diagram, describes the organization and wiring of the physical **c**omponents in a system. Component diagrams are often drawn to help model implementation details and double-check that every aspect of the system's required function is covered by planned development.

5.1.8 ER Diagram:

An Entity–relationship model (ER model) describes the structure of a database with the help of a diagram, which is known as Entity Relationship Diagram (ER Diagram). An ER model is a design or blueprint of a database that can later be implemented as a database. The main components of E-R model are: entity set and relationship set.

An ER diagram shows the relationship among entity sets. An entity set is a group of similar entities and these entities can have attributes. In terms of DBMS, an entity is a table or attribute of a table in database, so by showing relationship among tables and their attributes, ER diagram shows the complete logical structure of a database. Let’s have a look at a simple ER diagram to understand this concept.

5.1.9 DFD Diagram:

A Data Flow Diagram (DFD) is a traditional way to visualize the information flows within a system. A neat and clear DFD can depict a good amount of the system requirements graphically. It can be manual, automated, or a combination of both. It shows how information enters and leaves the system, what changes the information and where information is stored. The purpose of a DFD is to show the scope and boundaries of a system as a whole.

It may be used as a communications tool between a systems analyst and any person who plays a part in the system that acts as the starting point for redesigning a system.

5.2 SYSTEM ARCHITECTURE

Architecture diagrams can help system designers and developers visualize the high-level, overall structure of their system or application for the purpose of ensuring the system meets their users' needs. They can also be used to describe patterns that are used throughout the design. It's somewhat like a blueprint that can be used as a guide for the convenience of discussing, improving, and following among a team.

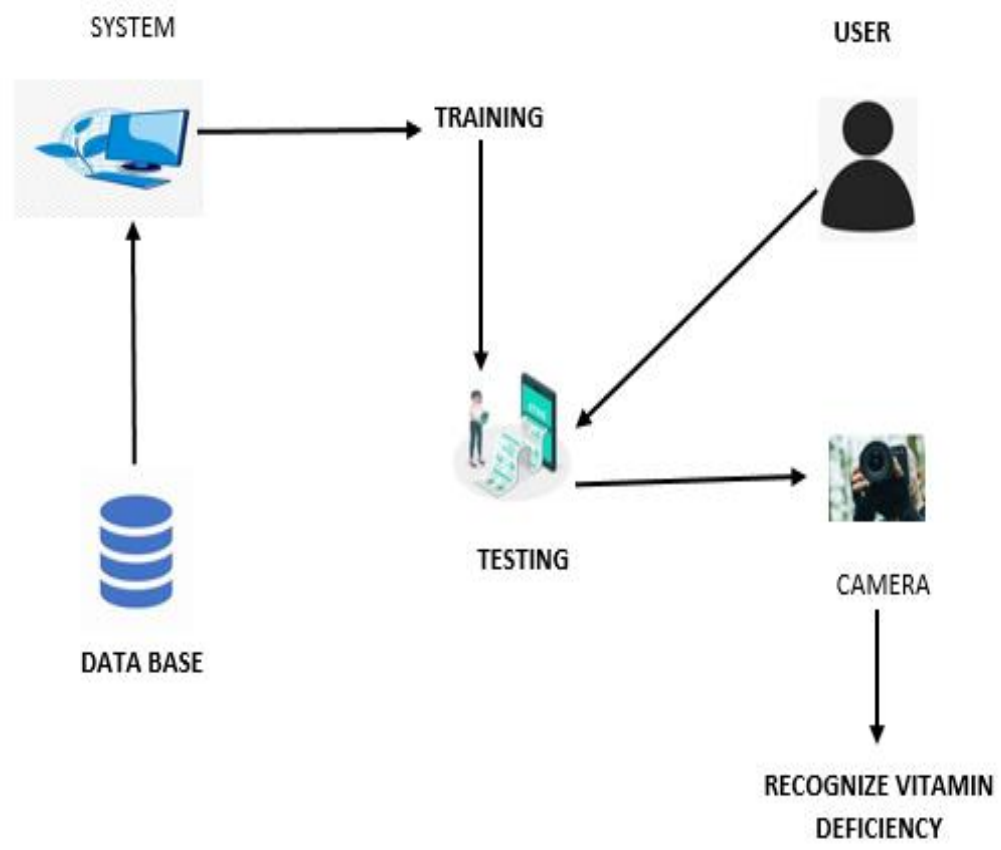

Fig 5.6: System Architecture

5.3 Flowchart

A flowchart is simply a graphical representation of steps. It shows steps in sequential order and is widely used in presenting the flow of algorithms, workflow or processes. Typically, a flowchart shows the steps as boxes of various kinds, and their order by connecting them with arrows. It originated from computer science as a tool for representing algorithms and programming logic but had extended to use in all other kinds of processes. Nowadays, flowcharts play an extremely important role in displaying information and assisting reasoning. They help us visualize complex processes, or make explicit the structure of problems and tasks. A flowchart can also be used to define a process or project to be implemented.

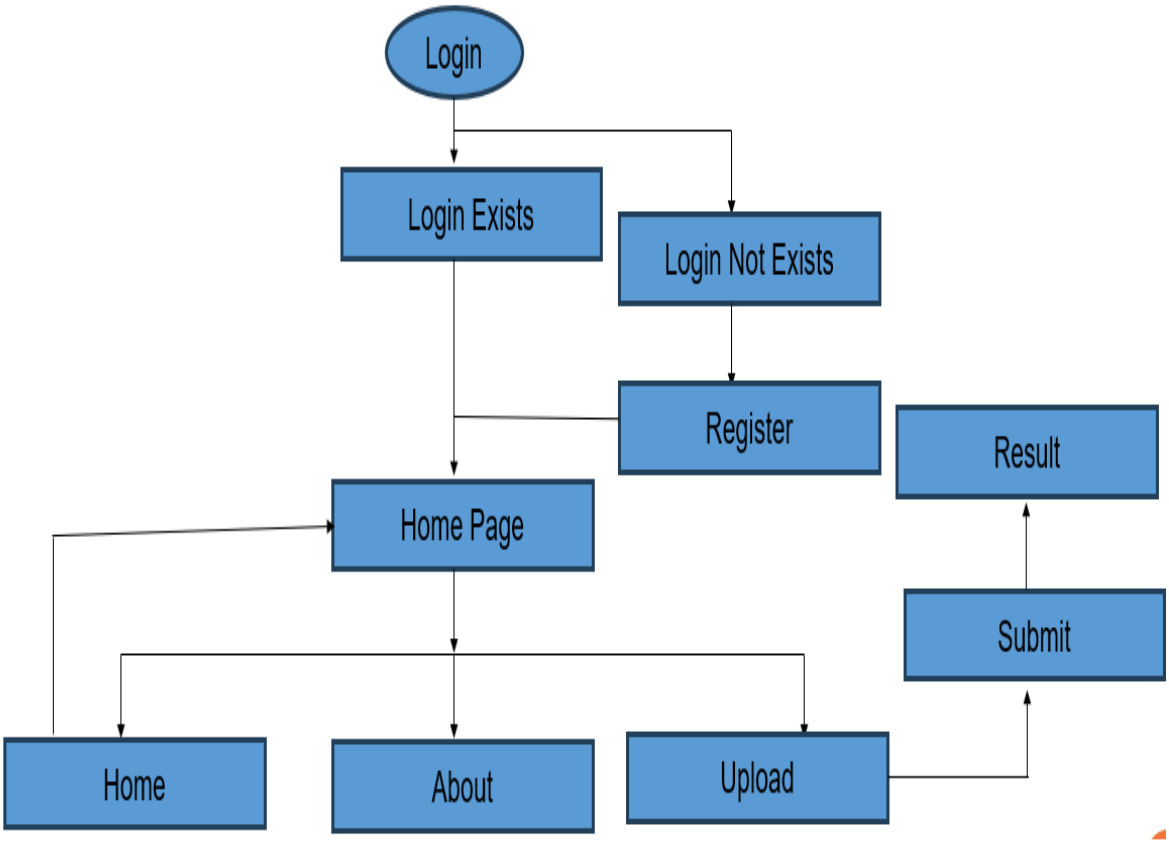

Fig 5.7: Flowchart of the system

CHAPTER 6

IMPLEMENTATION

Work flow of Proposed System

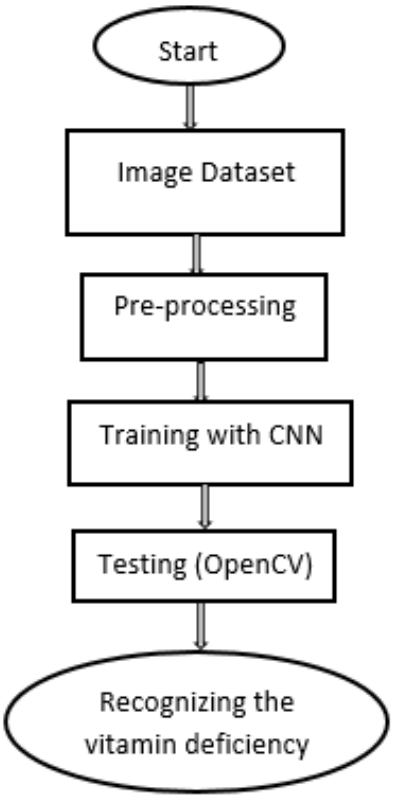

Fig 6.1: Block diagram of proposed method

Introduction of Input Design:

Input Design

The input design focuses on how data is collected, processed, and prepared for use by the system. Key aspects include:

- **Data Collection Interface:**
  - User-friendly interfaces for data entry, which may include forms or questionnaires for patients to input dietary information, symptoms, lifestyle factors, and medical history.
  - Options for healthcare professionals to input or upload clinical data, like physical examination results.
- **Image and Signal Input (if applicable):**

- Integration of modules to upload and process images (like retinal scans) or signals (like ECG data) that could indicate vitamin deficiencies.

- **Data Preprocessing:**
  - Automated checks for data completeness and validity.
  - Normalization and standardization of data to ensure consistency.
  - Secure data encryption during transmission to the processing server.
- **Integration with External Systems:**
  - Options for importing data from electronic health records (EHR) or other healthcare databases.
  - Compatibility with various data formats and standards used in healthcare.
- **Mobile and Web Applications:**
  - Responsive design for various devices, enabling data entry through mobile apps or web portals.
  - Feature to capture real-time data like dietary logs through mobile devices.

OUTPUT DESIGN

The output design focuses on how the results of the data analysis are presented to the users. Key aspects include:

- **Diagnostic Reports:**
  - Clear and concise reports indicating the likelihood or presence of vitamin deficiencies.
  - Visual aids like graphs or charts to illustrate the findings.
- **Recommendations and Alerts:**
  - Personalized recommendations based on the analysis, such as dietary suggestions.
  - Alerts or warnings for severe deficiency indications that require immediate medical attention.
- **User Dashboard:**
  - Interactive dashboards for users to view their health analysis over time.
  - Options for users to drill down into specific aspects of their health data.
- **Export and Sharing Options:**
  - Functionality to export reports in various formats (PDF, Excel, etc.) for sharing with healthcare providers.
  - Secure sharing options that comply with data privacy regulations.

- **Notifications and Reminders:**
  - Automated notifications to users for follow-up tests or check-ups.
  - Reminders for regular data input, such as updating dietary logs.
- **Data Visualization:**
  - Advanced data visualization tools for healthcare professionals to analyze trends and patterns in patient data.
  - Graphical representations to aid in understanding complex data insights.
- **Accessibility Features:**
  - Output design that is accessible to users with disabilities, including screen reader compatibility and high-contrast modes.

6.1 Datasets:

Machine Learning depends heavily on data. It‘s the most crucial aspect that makes algorithm training possible. It uses historical data and information to gain experiences. The better the collection of the dataset, the better will be the accuracy.

At the time of voter registration system going to capture 200 images through webcam and that is going to store in the folder with generated pin. This dataset can be used to train the model.

The dataset containing images of human parts with vitamin deficiency which are to be classified is split into training and testing dataset with the test size of 30-20%.

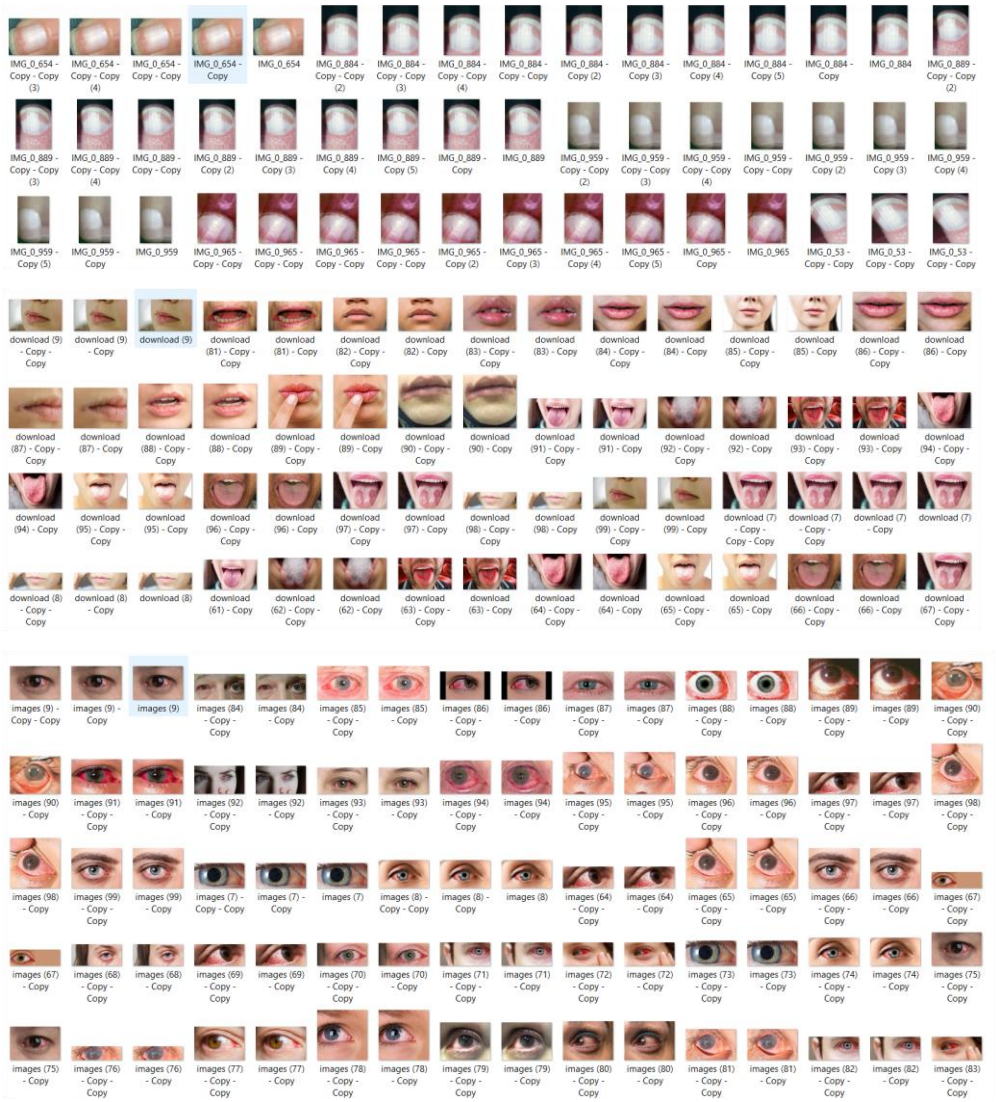

Fig 6.2: Dataset Collection

6.2 Data Pre-processing:

Resizing and reshaping the images into appropriate format to train our model. Data Pre-Processing is a Data Mining method that entails converting raw data into a format that can be understood. Real-world data is frequently inadequate, inconsistent, and/or lacking in specific activities or trends, as well as including numerous inaccuracies. This might result in low-quality data collection and, as a result, low-quality models based on that data. Preprocessing data is a method of resolving such problems. Machines do not comprehend free text, image, or video data; instead, they comprehend 1s and 0s. So putting on a slideshow of all our photographs and expecting our machine learning model to learn from it is probably not going to be adequate. Data Pre-processing is the step in any Machine Learning process in which the data is changed, or encoded, to make it easier for the machine to parse it.

In other words, the algorithm can now easily interpret the data's features. Data Pre-processing can be done in four different ways. Data cleaning/cleaning, data integration, data transformation, and data reduction are the four categories.

6.3 Training:

Use the pre-processed training dataset is used to train our model using CNN algorithm.

6.4 Classification:

The results of our model is display of classified vitamin deficiency.

Sure, here are the key points summarizing NasNetMobile's operation in the context of vitamin deficiency detection:

- CNN Architecture: NasNetMobile is a convolutional neural network (CNN) architecture specifically designed for image classification tasks.
- Image Analysis: It operates by analyzing images, such as photographs of individuals or specific body parts affected by deficiency symptoms.
- Feature Extraction: NasNetMobile extracts relevant features from these images, which may include skin color, texture, and other visual cues associated with different types of deficiencies.
- Training Data: It requires a dataset containing images labeled with corresponding deficiency types for training. The efficacy of NasNetMobile depends on the quality and diversity of this training data.
- Learning Patterns: Through training, NasNetMobile learns to recognize patterns indicative of various deficiencies from the labeled images in the dataset.
- Inference Process: During inference, the algorithm processes new images and classifies them based on the learned patterns, thereby identifying potential deficiencies in individuals.
- Efficacy Factors: The efficacy of NasNetMobile in this task is influenced by factors such as the quality and diversity of the training data, as well as the network's architecture and training parameters.

CHAPTER 7

TESTING

Software testing is an investigation conducted to provide stakeholders with information about the quality of the software product or service under test. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include the process of executing a program or application with the intent of finding software bugs (errors or other defects), and verifying that the software product is fit for use.

Software testing involves the execution of a software component or system component to evaluate one or more properties of interest. In general, these properties indicate the extent to which the component or system under test:

- Meets the requirements that guided its design and development,
- Responds correctly to all kinds of inputs,
- Performs its functions within an acceptable time,
- It is sufficiently usable,
- Can be installed and run in its intended environments, and
- Achieves the general result its stakeholder‘s desire.

7.1 Functionality Testing

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

- Database connection is successfully established.
- The flow of the application from one page to another is correct, accurate and quick.
- All the forms included in the application are working as expected.
- Proper alert messages are displayed in case of wrong inputs.
- After every action on the application the appropriate data is fetched from the backend.

7.2 Usability Testing

- The application enables smooth navigation, hence gives a user-friendly experience.
- The inputs taken from the user are via dropdown hence correct inputs are provided to the system.
- Wrong inputs given by the system are handled effectively.
- The content provided by the application is verified and is taken by the trusted sources.
- The datasets trained for prediction of the crop yield are accurate and balanced.

7.3 Interface Testing

- The application connects correctly with the server. In case of failure an appropriate message is displayed.
- Interruptions by the server or by the user are handled efficiently.
- If the user enters wrong credentials or invalid email id, the application handles it efficiently by displaying appropriate messages.
- The interaction with the user is smooth and easy.

7.4 Performance Testing

- It works fine without internet.
- The connection is secured and user details are stored in a secured manner.
- The switch from one screen to another is quick and smooth.
- The inputs from users are taken correctly and response is recorded quickly.

7.5 Unit Testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs.

All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform Crop Yield Prediction and Fertilizer Analysis Using Machine Learning basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach
Field testing will be performed manually and functional tests will be written in detail.
Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed

All links should take the user to the correct page

7.6 Integration Testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

7.7 Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

7.8 System Testing

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the

Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

SYSTEM STUDY AND TESTING

Feasibility Study:

The feasibility of the project is analysed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed

system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Economic feasibility:

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus, the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

Technical feasibility:

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client.

Social feasibility:

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

7.9 White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

7.10 Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box . you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

CHAPTER 8

RESULTS

In the final implementation of the application the first screen the user can view is the Login page that contain Email, Password text boxes and Login button. If the new user has to login, then they have to “Sign Up”.

Login:

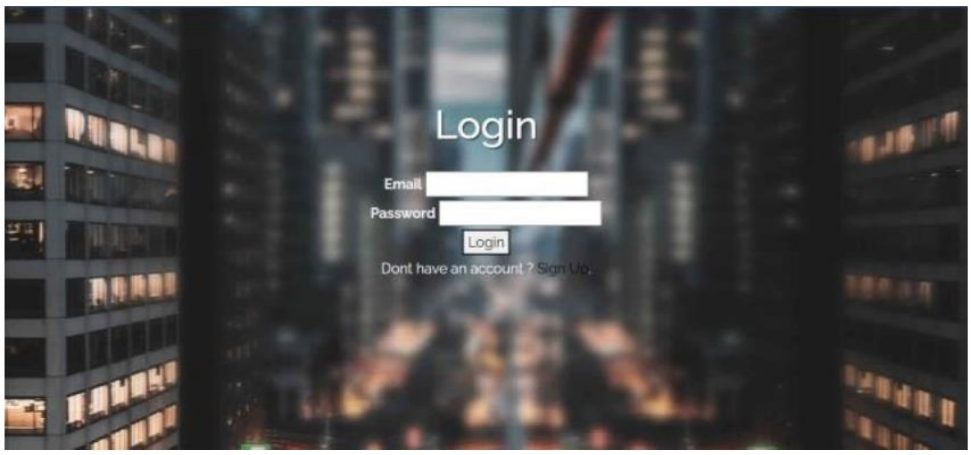

Fig 8.1: Login Page

The above figure represents the Initial web page of the system that has four functionalities that are User Registration, User Login Details. On selecting the required one we get the new webpage and required to give the inputs.

Register:

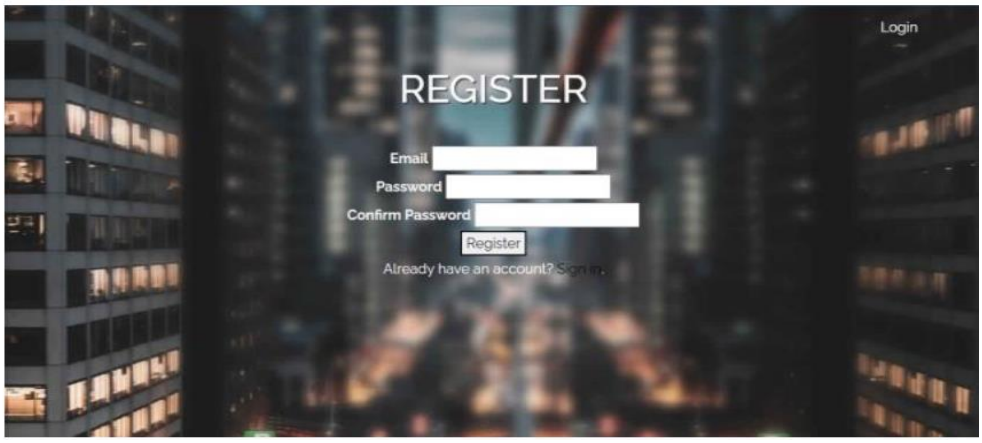

Fig 8.2 : Registration Page

The above page represents the initial page of the user registration. Here we are required to give inputs of Email, Password, Confirm Password. Then user has to click on “Register” button

Home:

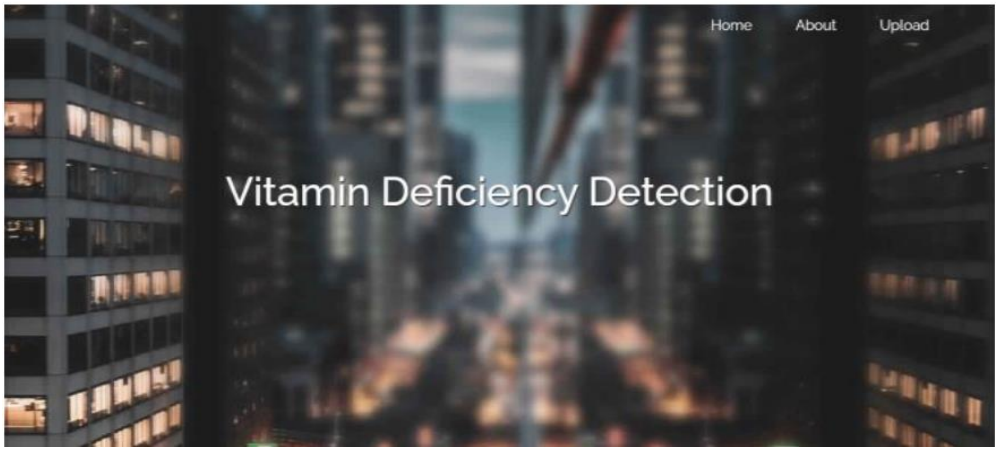

Fig 8.3: Home Page

The above page is the second page of login. Here we have three options as “Home”, “About”, “Upload”. If the user wants to go back to home page need to select “Home” option. If the user wants to know about the project need to select “About” option. Here we need to select “Upload” option to upload an image.

Upload

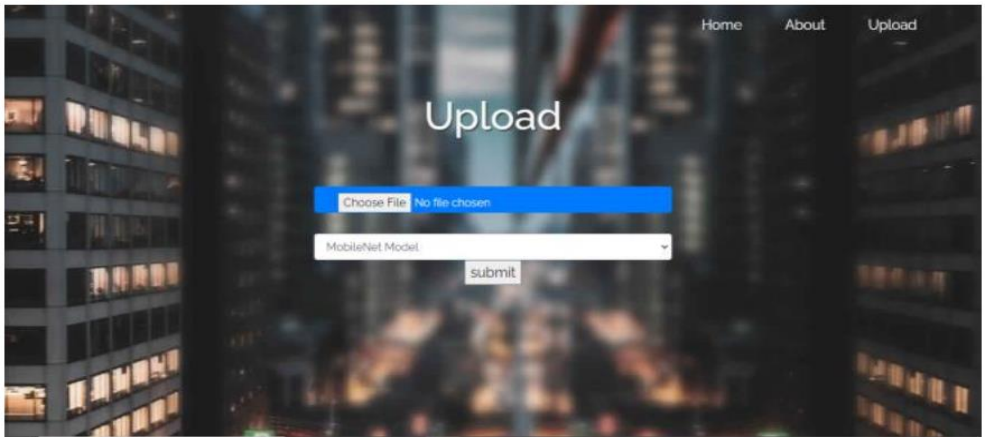

Fig 8.4: Upload Page

The above page is the third page of Login. Images of the victim can be taken through system and uploaded .

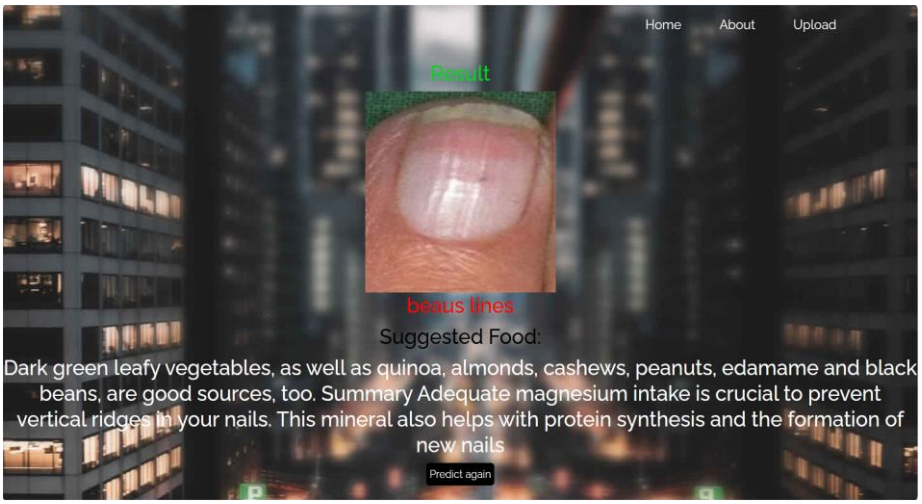

Fig 8.5: Result Page

The above page is the final page of login. Here we are supposed to get desired output.

CONCLUSION

In our proposed model we made the predictions of vitamin deficiency using Convolution Neural Network (CNN) from the deep learning and with the help of OpenCV. We considered a dataset of eye, lips, tongue and nails and trained using the CNN algorithm of deep learning. Once the training is completed, we used OpenCV that which recognizes and make predictions of vitamin deficiency. Vitamin deficiency detection using image processing and CNNs offers several advantages over existing systems. By leveraging the power of computer vision and deep learning, the proposed system provides a non-invasive, accurate, and accessible approach to detect and monitor nutrient deficiencies. Compared to traditional methods such as blood tests and clinical evaluations, the proposed system eliminates the need for invasive procedures, making it more convenient and comfortable for individuals. Additionally, the system has the potential for widespread implementation as it can utilize readily available imaging devices, including smartphones, reducing the cost and time associated with specialized equipment and laboratory tests. The use of image processing techniques and CNNs allows for automatic analysis of visual cues in various body parts, enabling the detection of multiple types of nutrient deficiencies. This broadens the scope of detection beyond individual nutrients and offers a comprehensive assessment of nutritional status. Furthermore, the proposed system provides real-time detection and monitoring, allowing for early intervention and personalized recommendations. By detecting deficiencies at an early stage, individuals can make informed dietary adjustments or receive appropriate supplementation, potentially preventing the progression of related health complications. In terms of accuracy, the proposed system harnesses the capabilities of CNNs for robust feature extraction and classification. By training the model on a diverse dataset of individuals with known deficiencies, the system achieves high accuracy in identifying different types of nutrient deficiencies

Future Work

This process can be extended in future to classify the more types of predictions of different classifications and we can also use the different types of transfer learning algorithms for better predictions.

REFERENCES

[1] Adult blindness secondary to vitamin A deficiency associated with an eating disorder. [online] Available at: https://www.ncbi.nlm.nih.gov/pubmed/15850971.

[2] Vitamin a deficiency and clinical disease: an historical overview. [online] Available at: https://www.ncbi.nlm.nih.gov/pubmed/18806089.

[3] Investing in the future: A united call to action on vitamin and mineral deficiencies. Global Report 2009.

[4] "92% of U.S. Population Have Vitamin Deficiency. Are You One of Them? - the biostation Delray Beach & Miami", the biostation Delray Beach & Miami, 2019. [Online]. Available: https://thebiostation.com/bioblog/nutrient-iv-therapy/do-you-havevitamin-deficiency/. [Accessed: 28-Jul- 2019].

[5] Glossitis with linear lesions: an early sign of vitamin B12 deficiency. [online] Available at: https://www.ncbi.nlm.nih.gov/pubmed/19231648.

[6] [Clinical manifestations of the mouth revealing Vitamin B12 deficiency before the onset of anemia]. [online] Available at: https://www.ncbi.nlm.nih.gov/pubmed/12671582.

[7] Atrophic glossitis from vitamin B12 deficiency: a case misdiagnosed as burning mouth disorder. [online] Available at: https://www.ncbi.nlm.nih.gov/pubmed/17209796.

[8] Sharecare. (2019). How does vitamin deficiency affect oral health? | Vitamins. [online] Available at: https://www.sharecare.com/health/vitamins-supplements/vitamindeficiency-affect-oral-health [Accessed 28 Jul. 2019].

[9] "Angular Cheilitis, Part 1: Local Etiologies" (PDF). www.skinandallergynews. Archived from the original (PDF) on 2013- 12-16. Retrieved 2014-04-21.

[10] Podiatry Today. (2019). When Vitamin And Nutritional Deficiencies Cause Skin And Nail Changes. [online] Available at: https://www.podiatrytoday.com/when-vitamin-and-nutritionaldeficiencies-cause-skin-and-nail-changes [Accessed 28 Jul. 2019].

[11] Healthline. (2019). Nail Abnormalities: Symptoms, Causes, and Prevention. [online] Available at: https://www.healthline.com/health/nail-abnormalities-2 [Accessed 28 Jul. 2019].

[12] Wollina U, Nenoff P, Haroske G, Haenssle HA (July 2016). "The Diagnosis and Treatment of Nail Disorders". Deutsches Arzteblatt International. 113 (29–30): 509–18. doi:10.3238/arztebl.2016.0509. PMC 5527843. PMID 27545710.

[13] Ben-Dayan D, Mittelman M, Floru S, Djaldetti M (1994). "Transverse nail ridgings (Beau's lines) induced by chemotherapy". Acta Haematol. 91 (2): 89–90. doi:10.1159/000204261. PMID 7517608.

[14] Park J; Li K (2010). "Multiple Beau's lines". New England Journal of Medicine. 362 (20): e63. doi:10.1056/NEJMicm0906698. PMID 20484394.

[15] Precision Nutrition. (2019). All About Nutrient Deficiencies. [online] Available at: https://www.precisionnutrition.com/aa-nutrientdeficiencies [Accessed 28 Jul. 2019].

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:03/March-2024 Impact Factor- 7.868 www.irjmets.com

NEURAL NETWORK BASED VITAMIN DEFICIENCY DETECTION USING IMAGES

Shajahan P*1, Pallavi R*2, Sai kiran M*3, Vamsi C*4, Sreekar Vamsi Krishna G*5

*1Assistant Professor, Department Of Computer Science And Engineering, Srinivasa Ramanujan Institute Of Technology, Anantapur, India.

*2,3,4,5Student, Department Of Computer Science And Engineering, Srinivasa Ramanujan Institute Of Technology, Anantapur, India.

DOI : https://www.doi.org/10.56726/IRJMETS50799

ABSTRACT

Vitamin deficiencies pose a significant global health concern, resulting in various debilitating conditions. Detecting these deficiencies early on is crucial for effective intervention and prevention. This study presents a novel technique that uses image processing and neural networks to identify vitamin deficiencies. The system utilizes image analysis techniques to evaluate physical signs of deficiencies, such as abnormalities in the skin, nails, or eyes. Detailed images of these symptoms are collected and processed to extract relevant features. A convolutional neural network (CNN) is then utilized to classify these features and identify potential deficiencies. The neural network is trained on a comprehensive dataset of images representing different vitamin deficiencies, ensuring accurate and robust detection capabilities. To further enhance the system's performance, transfer learning and data augmentation techniques are employed. Experimental results demonstrate the system's effectiveness in accurately detecting vitamin deficiencies. Additionally, the non-invasive nature of image-based detection makes this approach convenient and accessible for widespread screening and monitoring, especially in resource-limited settings. In conclusion, this proposed methodology offers a promising solution for early detection and intervention of vitamin deficiencies, contributing to improved public health outcomes and overall well-being. This innovative approach bridges the gap between technology and healthcare, potentially revolutionizing the global fight against vitamin deficiencies.

Keywords: Vitamin Deficiency, Deep Learning, CNN.

I. INTRODUCTION

Vitamin deficiency is a significant health issue that arises from the failure to acquire the necessary spectrum of essential vitamins and minerals. Over 2 billion people worldwide suffer from vitamin deficiencies, with over 1.2 billion being zinc deficient and half a million dying each year. Iron deficiency causes anemia, resulting in over 100,000 deaths. In the UAE, over 90% of the population suffers from vitamin deficiencies. In the US, over 92% of the population suffers from at least one mineral or vitamin deficiency, despite the country not experiencing a starvation crisis. Nutrient-rich foods are often considered financially expensive, leading to nutrient-rich foods becoming more of a symbol of luxury rather than the standard of daily food intake.

Researchers have found that the soil itself is deficient in micronutrients, with the mineral content of vegetables like cabbage, lettuce, spinach, and tomatoes depleting from 400 milligrams to less than 50 milligrams. Even with a perfect diet, something is missing, with 50% of Americans deficient in vitamin A, vitamin C, and magnesium, 70% of elderly Americans, and 90% of Americans of color being vitamin D deficient. A survey of 100 university students found 67% of them unaware of having vitamin deficiency.

Approximately 2 billion people worldwide suffer from vitamin deficiencies; of these, more than 1.2 billion are zinc deficient, and each year, half a million of them die. Every year, an iron deficiency-related anemia claims the lives of about 100,000 people. Insufficient intake of vitamins affects about 90% of the population in the United Arab Emirates. More than 92% of Americans have at least one vitamin or mineral deficiency, even in the absence of a famine. Nutrient-rich foods are becoming less and less essential to a person's regular diet because of their high perceived cost and more of a status symbol.

Over two billion people worldwide suffer from vitamin insufficiency, with one in three children not receiving enough vitamins. Vitamin A deficiency affects 33% of young children under five, leading to low immunity and night blindness. Vitamin deficits often coexist with mineral shortages, with children and pregnant women being

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:03/March-2024 Impact Factor- 7.868 www.irjmets.com

the most at risk. Common deficiencies include vitamin A, vitamin B, folate, and vitamin D. Supplementation programs have made diseases like scurvy and pellagra rare.

Vitamin deficiencies cause numerous health problems, as they hinder the intake of essential minerals and nutrition. It is challenging to measure nutritional requirements without medical advice, especially when people lack knowledge of the specific type of shortage they may be experiencing. Over 1.2 billion people worldwide suffer from zinc deficiency, with 500,000 passing away annually. Iron deficiency causes anemia that kills over 100,000 individuals annually.

In the UAE, vitamin deficiencies affect over 90% of the population. In America, over 92% of the population has at least one mineral or vitamin deficit. Nutrient-rich foods have become more of a luxury due to the availability of inexpensive, easily accessible junk foods. Researchers have discovered micronutrient deficiencies in soil, with magnesium, vitamin A, and vitamin C deficiencies affecting 50% of Americans, and vitamin D deficiency affecting 90% of Americans of color and 70% of older Americans.

II. RELATED WORK

A lack of vitamin A can cause a number of clinical symptoms, including growth abnormalities, xerophthalmia, and an increased risk of serious infections. Its existence has been known since antiquity, when experimental animal research and epidemiological observations conducted in the early 1900s confirmed its presence. The extent and effects of vitamin A insufficiency on the impoverished in low- and middle-income nations have been shown in recent clinical investigations, which have altered international health policy. Numerous stories have confirmed these results, underscoring the significance of treating vitamin A insufficiency. An early clinical indicator of cobalamin deficiency is glossitis with linear lesions, as reported in four cases without anemia or neurologic signs. Even in patients without anemia, measuring cobalamin levels is advised. Potential applications of image-based analysis for nutritional assessments are demonstrated by a machine learning system that uses facial photos to accurately detect vitamin D insufficiency. Using images of the tongue, researchers created a deep convolutional neural network architecture that can identify vitamin B12 deficiency with high accuracy, differentiating between healthy and deficient subjects.

III. MATERIAL AND METHODOLOGY

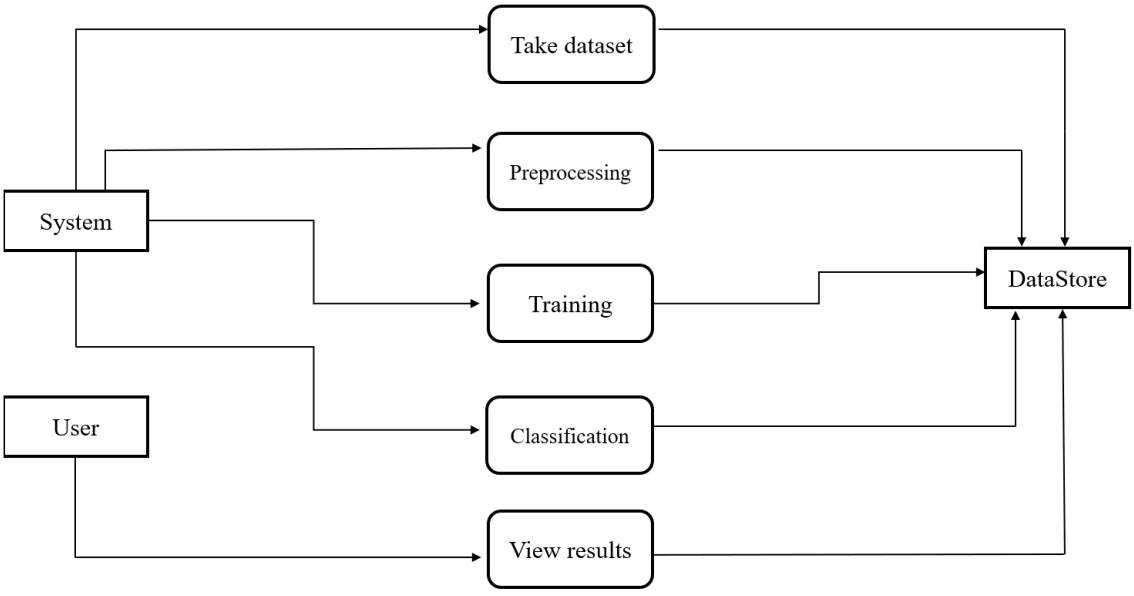

Fig 1: Data Flow Diagram

NasNetMobile is a convolutional neural network (CNN) architecture developed by Google's DeepMind. It's designed for image classification tasks, meaning it's trained to analyze and categorize images into different classes. In the context of detecting vitamin deficiency, NasNetMobile can be utilized as follows:

A. Data Collection: First, a dataset of images related to symptoms or indicators of various vitamin deficiencies is compiled. These images could include pictures of skin conditions, nails, eyes, or other relevant body parts affected by different deficiencies.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:03/March-2024 Impact Factor- 7.868 www.irjmets.com

B. Preprocessing: The preprocessing of dossier necessary handling gone dossier and excessive data . Missing dossier was controlled by first removing cases with important gone principles for most of the features. Instances accompanying any dossier features gone were suffused accompanying the most frequent happening profit each feature. Finally the structured dossier was the developing dataset holding 8000 images.

C. Training: NasNetMobile is trained on this dataset using a supervised learning approach. During training, the algorithm learns to recognize patterns and features in the images that are indicative of different types of deficiencies. For example, it may learn to associate certain skin discolorations or abnormalities with specific vitamin deficiencies.

D. Feature Extraction: The trained NasNetMobile model is then used to extract features from new images of individuals suspected of having vitamin deficiencies. These images could be captured using standard digital cameras or smartphones.

E. Classification: The extracted features are fed into the trained model, which then classifies the images based on the patterns it learned during training. The output of the classification process indicates the likelihood of the individual having a particular vitamin deficiency.

F. By leveraging NasNetMobile's capabilities in image classification, healthcare professionals can potentially streamline the process of identifying and diagnosing vitamin deficiencies, enabling earlier intervention and treatment. But it's vital to remember that even if NasNetMobile can help with the detection procedure, it should be used in conjunction with clinical expertise and other diagnostic methods for accurate assessment and diagnosis.

IV. ANALAZING ML ALGORITHM

One of the main aims of resolving different appliance knowledge algorithms was to find the best subspace of features unavoidable for vitamin deficiency detection. To do this, we completed activity all-encompassing data group, dossier combining, and development of forecasting models. We used following algorithms:

A. CNN:

A particular kind of deep learning model called a Convolutional Neural Network (CNN) is intended for processing and evaluating visual data, such as pictures. Its layers carry out convolution operations to extract features, such textures and edges, from the incoming data. These features are then passed through activation functions to introduce non-linearity, pooled to reduce spatial dimensions, and finally flattened into a vector. This vector is then processed by fully connected layers to produce the final output, which could be classifications or predictions. During training, the CNN adjusts its parameters through backpropagation and optimization algorithms to learn to recognize patterns and make accurate predictions. CNNs have become instrumental in tasks like image recognition, object detection, and classification, due to their ability to automatically learn hierarchical features from raw data.

B. ANN:

A computational model called an Artificial Neural Network (ANN) algorithm is modeled after the architecture and operation of biological neural networks, such as those found in the human brain. It is made up of layers of networked nodes called neurons. After receiving input signals and applying weights to them, each neuron uses an activation function to process the data and produce an output. An ANN's primary feature is its capacity to learn from data. During the training process, the network adjusts the weights between neurons to minimize prediction errors, effectively learning the underlying patterns and relationships within the data. ANN algorithms have found widespread application in various domains, including image and speech recognition, natural language processing, medical diagnosis, and financial forecasting. They excel at tasks involving pattern recognition, classification, regression, and clustering. Despite their power, ANNs require careful design, tuning, and training to achieve optimal performance, often involving experimentation with network architectures, activation functions, and optimization techniques.

C. InceptionRasnetV2:

The InceptionResNetV2 algorithm, a fusion of Inception and ResNet architectures, can be repurposed for detecting vitamin deficiencies through a process of fine-tuning and transfer learning. Initially trained on vast

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:03/March-2024 Impact Factor- 7.868 www.irjmets.com

datasets for image recognition, the model's adaptability allows it to learn specific features associated with symptoms of different vitamin deficiencies. By curating a diverse dataset of images displaying various deficiency indicators, such as pale skin or brittle nails, and preprocessing them for compatibility, the model undergoes training where it adjusts its parameters to minimize the disparity between predicted and actual deficiency types. Post-training evaluation ensures the model's accuracy and generalization, while deployment into practical settings like mobile apps or medical devices facilitates real-time deficiency detection. Continuous validation by domain experts is crucial for ensuring clinical relevance, and periodic updates may be necessary to accommodate new data and evolving understandings of vitamin deficiencies.

D. NasneMobilet:

NasNetMobile is a convolutional neural network (CNN) architecture designed for image classification tasks. In the context of vitamin deficiency detection, NasNetMobile likely operates by analyzing images, such as photographs of individuals or specific body parts affected by deficiency symptoms, and extracting relevant features from these images. These features could include skin color, texture, and other visual cues associated with different types of deficiencies. By training on a dataset containing images labeled with corresponding deficiency types, NasNetMobile learns to recognize patterns indicative of various deficiencies. During inference, the algorithm processes new images and classifies them based on the learned patterns, thereby identifying potential deficiencies in individuals. The efficacy of NasNetMobile in this task depends on the quality and diversity of the training data, as well as the network's architecture and training parameters.

Table 1: Outcomes of each Algorithm

Algorithms	Accuracy
ANN	68.8%
CNN	62.7%
Inception	83.33%
Nasnetmobile	97.7%

V. RESULT

Login:

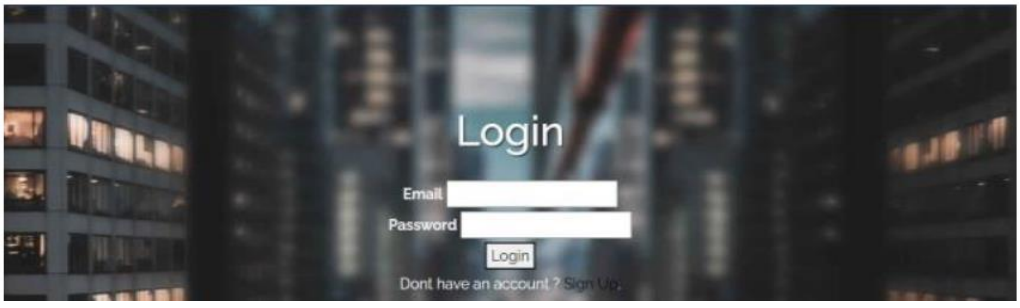

Fig 2: Login Page

Register:

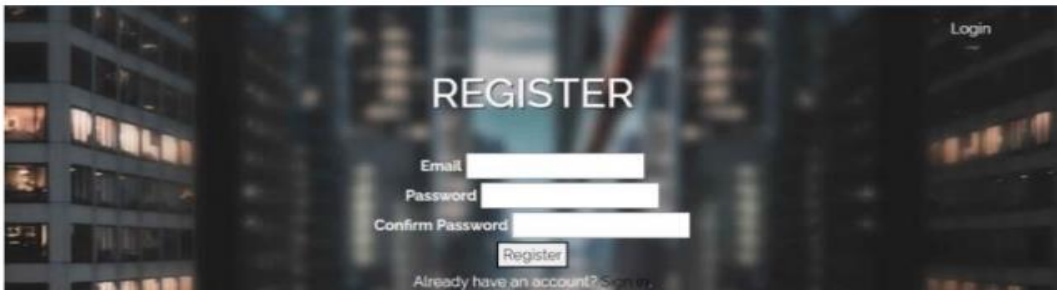

Fig 3: Registration Page

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:03/March-2024 Impact Factor- 7.868 www.irjmets.com

Home:

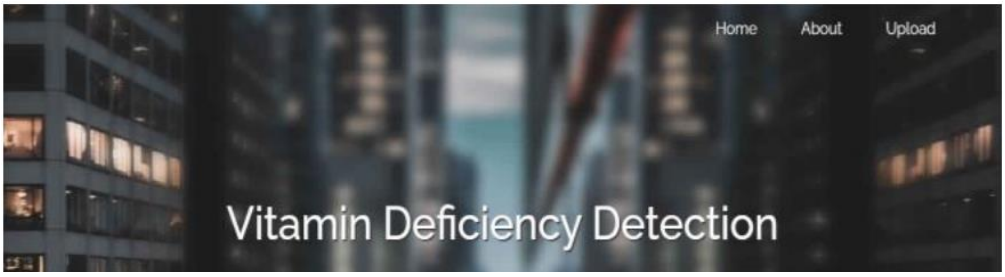

Fig 4: Home Page

Upload

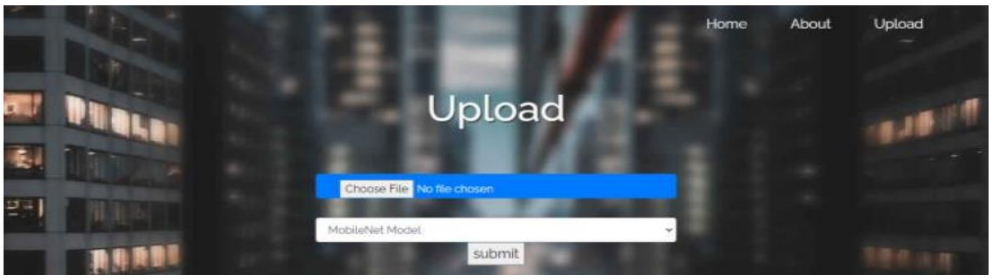

Fig 5: Upload Page

Fig 6: Result Page

VI. CONCLUSION

In summary, there are a number of benefits over current methods provided by the suggested project for vitamin insufficiency detection using image processing and CNNs. Through the utilization of computer vision and deep learning, the suggested system offers a non-invasive, precise, and easily accessible method for identifying and tracking nutritional deficits. The suggested approach is more convenient and comfortable for people because it does not require intrusive procedures, in contrast to conventional methods like blood testing and clinical evaluations.The technology also has the potential to be widely used because it can make use of widely accessible imaging devices, such as cellphones, which lowers the time and expense involved in specialized tools and laboratory testing. Multiple types of nutritional deficiencies can be detected by a computerized evaluation of visual cues in different body areas made possible by the use of CNNs and image processing techniques. This provides an in-depth analysis of nutritional status and extends the range of detection beyond specific nutrients.

Moreover, the suggested system offers real-time monitoring and detection, enabling early intervention and customized recommendations. Early detection of deficiencies allows people to take the right supplements or make educated dietary changes, which may slow the development of associated health issues. The suggested method leverages CNNs' extensive feature extraction and classification capabilities for high accuracy. The model is trained on a wide range of people with known weaknesses, which gives the system excellent accuracy in distinguishing between various vitamin deficiencies.

VII. REFERENCES

[1] Ahmed Saif Eldeen, Mohamed AitGacem, Saifeddin Alghlayini, Wessam Shehieb and Mustahsan Mir, "Vitamin Deficiency Detection Using Image Processing and Neural Network", IEEE Xplore, vol. 10, pp. 514-519, Jul.2020.

[2] Rutuja Moholar, Mansi Kamie, Gauri Bobade, Saijyoti Shinde, "Vitamin Deficiency Detection Using Image Processing and Neural Network", International Journal of Advance Research and Innovative Ideas in Education-International Standard Serial Number(IJARIIE-ISSN), vol. 9, pp. 4175-4182, Mar.2023.

[3] Khenilyn P. Lewis, Juancho D. Espineli, "Classification And Detection Of Nutritional Deficiencies In Coffee Plants Using Image Processing And Convolutional Neural Network (CNN)", InternationalJournalOf Scientific & Technology Research (IJSTR), vol. 9, pp. 2076-2081, Apr. 2020.

[4] Alfred Sommer, "Vitamin A Deficiency and Clinical Disease", The Journal of Nutrition, vol. 10, pp. 1835-1839, Feb.2018.

[5] Uwe Wollina, Pietro Nenoff, Gunter Haroske, Holger A. Haenssl, "The Diagnosis and Treatment of Nail Disorder", Deutsches Arzteblatt International , vol. 9, pp. 509-518, Mar.2016.

[6] https://www.nin.res.in/researchhighlights/2020-21.pdf"ICMR-National Institute of Nutrition 2020-21".

[7] Annadasu, P. and Jaisharma, K., 2020. Rice Quality Analysis Using Machine Learning (No. 4761). EasyChair.

[8] Mavalankar DV, Trivedi CC, Gray RH. Maternal weight, height and risk of poor pregnancy outcome in Ahmedabad, India. Indian Pediatr. 1994 Oct; 31(10):1205-12. PMID: 7875780.

[9] International Journal Of Scientific & Technology Research (IJSTR), vol. 9, pp. 2076-2081, Apr. 2020.

[10] International Journal of Advance Research and Innovative Ideas in Education-International Standard Serial Number(IJARIIE-ISSN), vol. 9, pp. 4175-4182, Mar.2023.

[11] Julia S. Lehman, Alison J. Bruce, Roy S. Rogers. " Atrophic Glossitis From Vitamin B Deficiency: A Case Misdiagnosed as Burning Mouth Disorder " , Journal of Periodontology, 2006.

[12] Referenced from https://bjp.sagepub.unboundmedicine.com

[13] Referenced from https://pure.johnshopkins.edu

[14] Alfred Sommer. "Vitamin A Deficiency and Clinical Disease: An Historical Overview" , The Journal of Nutrition, 2008.

[15] Based on Research papers Submitted to Universiti Kebangsaan Malaysia.

[16] Ahmed Saif Eldeen, Mohamed AitGacem, Saifeddin Alghlayini, Wessam Shehieb, Mustahsan Mir. "Vitamin Deficiency Detection Using Image Processing and Neural Network" , 2020 Advances in Science and Engineering Technology International Conferences (ASET), 2020.

[17] Dr. R. Maruthamuthu, T. Harika, "Vitamin Deficiency Detection Using Image Processing and Neural Network", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 9, Issue 4, pp.200-205, July-August-2023.

International Research Journal Of Modernization in Engineering Technology and Science

(Peer-Reviewed, Open Access, Fully Refereed International Journal)

e-ISSN: 2582-5208

Ref: IRJMETS/Certificate/Volume 06/Issue 03/60300087650

Date: 23/03/2024

Certificate of Publication

*This is to certify that author "**Shajahan P**" with paper ID "**IRJMETS60300087650**" has published a paper entitled "NEURAL NETWORK BASED VITAMIN DEFICIENCY DETECTION USING IMAGES" **in International Research Journal Of Modernization In Engineering Technology And Science (IRJMETS), Volume 06, Issue 03, March 2024***

Editor in Chief

IRJMETS Impact Factor 7.868

We Wish For Your Better Future

www.irjmets.com

International Research Journal Of Modernization in Engineering Technology and Science

(Peer-Reviewed, Open Access, Fully Refereed International Journal)

e-ISSN: 2582-5208

Ref: IRJMETS/Certificate/Volume 06/Issue 03/60300087650

Date: 23/03/2024

Certificate of Publication

This is to certify that author "**Pallavi R**" with paper ID "**IRJMETS60300087650**" has published a paper entitled "NEURAL NETWORK BASED VITAMIN DEFICIENCY DETECTION USING IMAGES" **in International Research Journal Of Modernization In Engineering Technology And Science (IRJMETS), Volume 06, Issue 03, March 2024**

Editor in Chief

IRJMETS Impact Factor 7.868

We Wish For Your Better Future

www.irjmets.com

International Research Journal Of Modernization in Engineering Technology and Science

(Peer-Reviewed, Open Access, Fully Refereed International Journal)

e-ISSN: 2582-5208

Ref: IRJMETS/Certificate/Volume 06/Issue 03/60300087650

Date: 23/03/2024

Certificate of Publication

*This is to certify that author "**Sai kiran M**" with paper ID "**IRJMETS60300087650**" has published a paper entitled "NEURAL NETWORK BASED VITAMIN DEFICIENCY DETECTION USING IMAGES" **in International Research Journal Of Modernization In Engineering Technology And Science (IRJMETS), Volume 06, Issue 03, March 2024***

Editor in Chief

IRJMETS Impact Factor 7.868

We Wish For Your Better Future

www.irjmets.com

International Research Journal Of Modernization in Engineering Technology and Science

(Peer-Reviewed, Open Access, Fully Refereed International Journal)

e-ISSN: 2582-5208

Ref: IRJMETS/Certificate/Volume 06/Issue 03/60300087650

Date: 23/03/2024

Certificate of Publication

*This is to certify that author "**Vamsi C**" with paper ID "**IRJMETS60300087650**" has published a paper entitled "NEURAL NETWORK BASED VITAMIN DEFICIENCY DETECTION USING IMAGES" **in International Research Journal Of Modernization In Engineering Technology And Science (IRJMETS), Volume 06, Issue 03, March 2024***

Editor in Chief

IRJMETS Impact Factor 7.868

We Wish For Your Better Future

www.irjmets.com

International Research Journal Of Modernization in Engineering Technology and Science

(Peer-Reviewed, Open Access, Fully Refereed International Journal)

e-ISSN: 2582-5208

Ref: IRJMETS/Certificate/Volume 06/Issue 03/60300087650

Date: 23/03/2024

Certificate of Publication

*This is to certify that author "**Sreekar Vamsi Krishna G**" with paper ID "**IRJMETS60300087650**" has published a paper entitled "NEURAL NETWORK BASED VITAMIN DEFICIENCY DETECTION USING IMAGES" **in International Research Journal Of Modernization In Engineering Technology And Science (IRJMETS), Volume 06, Issue 03, March 2024***

Editor in Chief

IRJMETS Impact Factor 7.868

We Wish For Your Better Future

www.irjmets.com