

HIDDEN HUNGER DETECTION USING DEEP LEARNING APPROACHES

A PROJECT REPORT

Submitted by

LAVANYA L	953219106016
PANDI MEENA R	953219106023
PONCHENDILA N	953219106024
SUBASHINI P	953219106038

In partial fulfillment for the award of the degree

Of

BACHELOR OF ENGINEERING

IN

ELECTRONICS AND COMMUNICATION ENGINEERING



UNIVERSITY VOC COLLEGE OF ENGINEERING

THOOTHUKUDI-628008

ANNA UNIVERSITY : CHENNAI 600 025

MARCH 2023

ANNA UNIVERSITY : CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report “**HIDDEN HUNGER DETECTION USING DEEP LEARNING APPROACHES**” is the bonafide work of

LAVANYA L	953219106016
PANDI MEENA R	953219106023
PONCHENDILA N	953219106024
SUBASHINI P	953219106038

who carried out the project work under my supervision.

SIGNATURE
Dr.K.ESAKKI MUTHU,M.E.,Ph.D.,

HEAD OF THE DEPARTMENT

Department of Electronics and
Communication Engineering,
University VOC College of Engg,
Thoothukudi-628008

SIGNATURE
Dr.A.MOOKAMBIGA,M.Tech.,Ph.D.,

SUPERVISOR

Department of Electronics and
Communication Engineering,
University VOC College of Engg,
Thoothukudi-628008

Submitted for the project viva-voce examination held on_____

INTERNAL EXAMINER

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

It is a difficult task in life to choose words to express one's gratitude towards the beneficiaries. We are very grateful to the **ALMIGHTY GOD** who helped us the way throughout the project and who has modeled us into what we are today.

We wish to express our heartfelt regards and sincere thanks to **Dr.C.PETER DEVADOSS,M.E.,Ph.D.**, Dean ,University VOC College of Engineering for his constant encouragement during the course of this project work.

We wish to express our thanks to our Head of the Department and Guide **Dr.K.ESAKKI MUTHU,M.E.,Ph.D.**, Assistant Professor, Department of Electronics and Communication Engineering for his stimulating guidance, continuous encouragement and supervision throughout the course of this work.

We express our sincere thanks and gratitude to our project Coordinator **Dr.A.MOOKAMBIGA,M.TECH.,Ph.D.**, Assistant Professor ,Department of Electronics and Communication Engineering, for her insightful comments and constructive suggestion to improve the quality of this project work.

We would like to place on record our deep sense of gratitude to **Dr.A.MOOKAMBIGA,M.TECH.,Ph.D.**, Assistant Professor ,Department of Electronics and Communication Engineering, for her generous guidance, help and useful suggestions.

We wish to express our deep sense of gratitude to our parents ,friends ,all our teaching and non-teaching staff members and all who participated enthusiastically with their constructive criticism either directly or indirectly in making this project a grand success.

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	vi
	LIST OF TABLE	vii
	LIST OF FIGURES	viii
	LIST OF ABBREVIATION	xii
1	INTRODUCTION	1
	1.1 THE HIDDEN HUNGER	1
	1.1.1 The steps taken to mitigate the hidden hunger	2
	1.2 NEED FOR THE STUDY	3
	1.3 OBJECTIVES	9
	1.4 ORGANIZATION OF CHAPTERS	9
2	LITERATURE REVIEW	10
	2.1 GENERAL	10
3	METHODOLOGY	17
	3.1 GENERAL	17
	3.2 BLOCK DIAGRAM	17
	3.3 INPUT DATASET	18
	3.4 DATA AUGMENTATION	18
	3.4.1 Image Augmentation Techniques	19
	3.5 SPLITTING THE DATASET	19
	3.6 FEATURE EXTRACTION USING DEEP LEARNING APPROACHES	20
	3.6.1 ResNet	22

	3.6.2 Inception V3	25
	3.6.3 Xception	25
	3.6.4 MobileNet V2	27
	3.6.5 Cnn model	29
	3.7 CLASSIFICATION USING DEEP LEARNING APPROACHES	30
	3.8 PERFORMANCE ANALYSIS	31
4	RESULTS AND DISCUSSION	34
	4.1 GENERAL	34
	4.2 HIDDEN HUNGER DETECTION IN HUMAN	34
	4.2.1 Data augmentation	37
	4.2.2 Splitting the dataset	39
	4.2.3 Deep Learning Models	39
	4.2.4 Image Prediction using deep learning models for nail dataset	55
	4.3 HIDDEN HUNGER DETECTION USING PLANT IMAGES	56
	4.3.1 Data Augmentation	59
	4.3.2 Splitting the dataset	61
	4.3.3 Deep learning models	61
	4.3.4 Image Prediction using deep learning models for leaf dataset	77
	4.4 WEB DEVELOPMENT	78
	4.4.1 Steps to access the webpage	78
5	SUMMARY AND CONCLUSION	80
	5.1 SUMMARY	80
	5.2 CONCLUSION AND FUTURE SCOPE	80
6	REFERENCES	82

ABSTRACT

Living Organisms need adequate amount of both macro and micro nutrients for complete and healthy life-cycle. Sufficient amount of micronutrients like Iron, Iodine, Zinc in human and Zn, Fe, Mn, Cu, Boron, Molybdenum in Plants are important for the normal and healthy growth. Lack or absence of micro nutrients leads to difficulties in carrying in the daily activities of plants and human. This micro nutrient deficiency that is not noticed at early stages is termed as Hidden Hunger which may cause serious health implications if left un-noticed. Thus, having the timely check for nutrient contents is important. Plants usually show the definite deficiency on their leaves with notable different patterns for each nutrient. In Human micro nutrient deficiency can be noticed through human nails which shows different patterns for definite deficiency. Our proposed work is to provide an automated and reliable economic solution for Hidden Hunger identification. The dataset is obtained from Kaggle and it is split into train, test, validation (as train data-70%, test data-20%, validation data-10%). For detecting micro nutrient deficiency in human we use healthy and nutrition deficient nail and eye images. The training dataset will be given to supervised machine learning for further detection and identification of exact nutrient deficiency in healthy crops and human in order to take preventive measures to attain zero hunger. The performance of developed model is evaluated in terms of accuracy, precision and sensitivity.

LIST OF TABLES

TABLE NO	TITLE	PAGE NO
1.1	Percentage of major micronutrient deficiency	4
1.2	Effect of micro nutrient deficiency in human	5
1.3	Effect of micronutrient deficiency in plants	7
4.1	Different classes in nail and eye dataset with their sample images	35
4.2	Data augmentation result	38
4.3	Augmentation Techniques	38
4.4	Comparative Analysis of accuracy and computational time of different models	49
4.5	Comparative Analysis of performance metrics	51
4.6	Different classes in leaf dataset and their sample images	57
4.7	Data augmentation result	59
4.8	Augmentation Techniques	60
4.9	Comparative Analysis of accuracy and computational time of different models	72
4.10	Comparative Analysis of performance metrics	73

LIST OF FIGURES

FIGURE NO	TITLE	PAGE NO
3.1	Block diagram of hidden hunger detection system	17
3.2	Convolution Operation	20
3.3	ReLU graph	22
3.4	Residual block	23
3.5	ResNet Architecture	24
3.6	Inception V3 Architecture	25
3.7	Xception Architecture	26
3.8	MobileNet V2 Architecture	28
3.9	Architecture of cnn model	29
3.10	Basic classification approach	30
3.11	Confusion matrix	32
4.1	Qualitative Analysis of ResNet 152 V2	41
4.2	Qualitative Analysis of Inception V3	43
4.3	Qualitative Analysis of Xception	45
4.4	Qualitative Analysis of MobileNet V2	47
4.5	Qualitative Analysis of cnn model	49
4.6	Qualitative Analysis of precision of each class	52

4.7	Qualitative Analysis of recall of each class	53
4.8	Qualitative Analysis of f1-score of each class	54
4.9	Qualitative Analysis of execution time of models	54
4.10	Image prediction using human datasets	56
4.11	Qualitative Analysis of ResNet 152 V2	63
4.12	Qualitative Analysis of Inception V3	65
4.13	Qualitative Analysis of Xception	67
4.14	Qualitative Analysis of MobileNet V2	69
4.15	Qualitative Analysis of cnn model	71
4.16	Qualitative Analysis of precision of each class	74
4.17	Qualitative Analysis of recall of each class	75
4.18	Qualitative Analysis of f1-score of each class	76
4.19	Qualitative Analysis of execution time of models	76
4.20	Image prediction using leaf Datasets	77

LIST OF ABBREVIATIONS

CNN	C onvolution N eural N etwork
GPU	G raphics P rocessing U nit
CPU	C entral P rocessing U nit
RAM	R andom A ccess M emory
VRAM	V ideo R andom A ccess M emory
GB	G iga B yte

CHAPTER 1

INTRODUCTION

1.1 THE HIDDEN HUNGER

The human body may only require micronutrients in small doses, but these vitamins and minerals are crucial for basic physiological functions such as metabolism, growth, and development. Deficiencies in one or more of these micronutrients may lead to detrimental health impacts, including chronic diseases. For decades, health organizations have been reporting on micronutrient malnutrition, including deficiencies in iodine, iron, folate, vitamin A, and zinc, among others, that continue to have devastating consequences for billions of people worldwide. The term “hidden hunger” is used to describe vitamin and mineral deficiency. When someone’s habitual diet consists of foods that lack necessary levels of micronutrients, the resulting health impacts may not always be acutely visible. Micronutrient malnutrition has been associated with a wide range of physiological impairments, including metabolic disorders; reduced immune, endocrine, and cognitive function; and delayed or inadequate physical development. Essential micronutrients human need are Iron, Vitamin- A, D, Iodine, Zinc, Folate (Vitamin- B9). There are 7 essential plant nutrient elements defined as micronutrients [boron (B), zinc (Zn), manganese (Mn), iron (Fe), copper (Cu), molybdenum (Mo), chlorine (Cl)]. They constitute in total less than 1% of the dry weight of most plants. Some micronutrient deficiencies may also be an underlying cause of chronic disease. Magnesium, for example, is present in a variety of foods, from greens to whole grains, yet it is under-consumed in the US. Studies indicate that low magnesium intake has been associated with a greater risk of several chronic diseases, including cardiovascular disease (CVD), type 2 diabetes, metabolic syndrome, depression, and impaired cognition. Although pregnant women, children and adolescents are often cited as the most vulnerable populations affected by

hidden hunger, it damages the health of people throughout the life cycle. Overcoming hunger and malnutrition in all its forms (including under nutrition, micronutrient deficiencies, overweight, and obesity) is about more than securing enough food to survive. A healthy diet prevents malnutrition and promotes well-being by ensuring we're consuming diverse meals full of essential fats, complex carbohydrates, vitamins, and minerals. Leading a healthy, adequate, and nutritious diet is essential to fight hidden hunger, but nutrition education also has a part to play in helping people make healthier choices. Ultimately, understanding the nutritional information available on food labels can empower people to make smarter eating decisions, choosing from a variety of foods and beverages that are higher in nutrient density throughout the day. Nutrient density is a measure of how much nutrition you get per serving or per calorie eaten. It's an important metric to develop a healthy diet. When choosing between two food items with the same calorie amount, nutrient dense food choice can provide your body with the protein, fiber, healthy fats, vitamins, and minerals we need every day, while low nutrient-dense choice may provide empty calories from sugar and saturated fat with no other significant nutrient.

1.1.1 The steps taken to mitigate the Hidden Hunger

There are three steps to overcome the hidden hunger problem. They are,

- a.** Supplementation
- b.** Fortification
- c.** Diet diversification.

These strategies are considered by the Food and Agricultural Organization (FAO) as sufficient in meeting the nutritional requirements of the general population

a. Supplementation: A technical approach, supplementations deliver nutrients directly to the population through syrup or pills. Its main advantage is that it can supply the specific number of nutrients in a highly absorbable form.

It should be noted that supplementation is a short term method eventually replaced by food-based measures in order to increase food and nutrient diversity.

Supplementation programs in India are focused on providing iron to pregnant women and vitamin A to children under 5.

b. Food fortification: Food fortification is the addition of micronutrients in processed foods. Enrichment and fortification are some of the methods used. Enrichment restores lost nutrients during food processing, while in fortification, those nutrients are added which are not originally found in the food being processed. Depending on the requirements, food fortifications is done to full-fill the following objectives

- Maintain nutritional quality of foods
- Prevent nutritional deficiencies in the population
- Provide technological functions in food processing

Food fortification plays a crucial role in enhancing nutritional improvement programmes and are considered as part of a hands-on approach to mitigate hidden hunger.

c. Diet Diversification: Through diet diversification the quantity and the range of micronutrients-rich foods can be increased. It requires different types of foods such as fruits, vegetables, pulses, dairy products etc to be readily available and that too in appropriate quantities. With diversity in diet, many food constituents like antioxidants, and probiotics can be taken, thus improving the nutrient intake of the population as such. Hence, diet diversification is the preferred method of combating hidden hunger. Several studies show that a higher dietary diversity is associated with better health and reduced risk of various chronic diseases. Not only the quantity of food but also the quality of diet has a significant impact on the overall health.

1.2 NEED FOR THE STUDY

Micronutrient deficiencies affect an estimated two billion people, or almost one-third of the world's population. Micro nutrient deficiency like Iron, Manganese, Zinc, Boron deficiency etc. are often noticed in plants and human. If these deficiencies are left untreated it may cause serious health issues in human. In plants

hidden hunger if not diagnosed may affect crop yield. Mostly macro nutrient deficiency detection in plants and human is highly focused and treated. A recent survey on hidden hunger describes that 30% of school children and people worldwide suffers from Vitamin D deficiency and 49% of crops across India suffer from Zinc deficiency. So it is extremely important to detect the micro nutrient deficiency at early stages. A Recent survey made across the world shows percentage of major micro nutrient deficiency.

Table 1.1: Percentage of major micro nutrient deficiency

CATEGORY	DEFICIENCY	PERCENTAGE (%)
Pre School Children	Vitamin- A	30
	Iron	18
People worldwide	Iodine	30
	Zinc	17

Table 1.1 Survey showing percentage of major micro nutrient deficiency across the world. Although micronutrients present in smaller quantity, it causes major chronic health issues in case of deficiency. The table shows that among preschool children 30% are affected by Vitamin A deficiency, 18% are affected by Iron deficiency and among people all over the world 30% are affected by Iodine deficiency, 17% are affected by Zinc deficiency. From this survey it is noticed that Preschool children are mainly affected by Vitamin A deficiency. So it is important to detect the micronutrient deficiency as early as possible. Early detection is extremely necessary so that the issues due to these micronutrient deficiency can be avoided.

Table 1.2: Effect of Micronutrient Deficiency in Human

S.NO.	MICRO NUTRIENT	EFFECT OF MALNUTRITION
1.	Iron (Fe)	Iron deficiency is a leading cause of anemia which is defined as low hemoglobin concentration. Anemia affects 40% of children younger than 5 years of age and 30% of pregnant women globally.
2.	Vitamin A	Children with vitamin A deficiency face an increased risk of blindness and death from infections such as measles and diarrhoea. Globally, vitamin A deficiency affects an estimated 190 million preschool-age children.
3.	Vitamin D	Vitamin D deficiency causes bone diseases, including rickets in children and Osteomalacia in adults. Vitamin D helps the immune system resist bacteria and viruses. Vitamin D is required for muscle and nerve functions.
4.	Iodine	Iodine is required during pregnancy and infancy for the infant's healthy growth and cognitive development. Globally an estimated 1.8 billion people have insufficient iodine intake. The American Thyroid Association and the American Academy of Pediatrics recommend that pregnant or breastfeeding women take a

		supplement every day containing 150 micrograms of iodine.
5.	Vitamin B12	Folate deficiency during pregnancy can cause severe complications. Folate is important for the growth of the fetus's brain and spinal cord. Folate deficiency can cause severe birth defects called neural tube defects.
6.	Zinc (Zn)	Zinc promotes immune functions and helps people resist infectious diseases including diarrhoea, pneumonia and malaria. Zinc is also needed for healthy pregnancies. Globally, 17.3% of the population is at risk for zinc deficiency due to dietary inadequacy; up to 30% of people are at risk in some regions of the world. Skin conditions associated with zinc deficiency include acrodermatitis enteropathica, cheilitis, and dermatitis.

The above table 1.2 shows that the effect of micronutrient deficiency in human due to the lack of Iron, Vitamin A, Vitamin D, Iodine, Vitamin B12, Zinc. The table describes what micronutrient deficiency causes what health issues. From this table it is inferred that even micronutrients present in small quantity in our body its deficiency may cause chronic health issues in our body. So it is essential to be aware of micronutrients proportion in our diet.

Table 1.3: Effect of Micronutrient Deficiency in Plants.

S.NO.	MICRO NUTRIENT	EFFECT OF MALNUTRITION
1.	Boron (B)	Boron deficiency affects vegetative and reproductive growth of plants resulting in inhibition of cell expansion, death of meristem and reduced fertility. Plants contain B both in a water-soluble and insoluble form.
2.	Zinc (Zn)	The physiological stress caused by Zn deficiency results in development of abnormalities in plants. In case of severe 'acute' Zn deficiency, visible symptoms include stunted growth, chlorosis of leaves, small leaves and spikelet sterility.
3.	Manganese (Mn)	In general, affected crops are pale green and growth is reduced. Specific symptoms may first appear on the youngest or oldest leaves and vary from species to species. The most common symptom is for leaves to turn pale green between the veins, with normal coloured areas next to the veins. As the deficiency progresses, the area between the veins becomes paler, enlarges and may brown and die.
4.	Iron (Fe)	The primary symptom of iron deficiency is interveinal chlorosis, the development of a yellow leaf with a network of dark green veins. In severe cases, the entire leaf turns

		yellow or white and the outer edges may scorch and turn brown as the plant cells die.
5.	Copper (Cu)	In most plants, young foliage is severely stunted as well as chlorotic. Deficient foliage can be cupped and deformed, bleached (lettuce), flaccid and blue green with chlorotic margins (tomato), abscise early (walnut), and eventually become necrotic in the interveinal areas.
6.	Molybdenum (Mo)	The main symptoms of molybdenum deficiency in non-legumes are stunting and failure of leaves to develop a healthy dark green color. The leaves of affected plants show a pale green or yellowish green color between the veins and along the edges. In advanced stages, the leaf tissue at the margins of the leaves dies. The older leaves are the more severely affected.
7.	Chlorine (Cl)	The most commonly described symptom of Cl deficiency is wilting of leaves, especially at the margins. As the deficiency progresses and becomes more severe, the leaves exhibit curling, bronzing, chlorosis, and necrosis.

The above table 1.3 shows that the effect of micronutrient deficiency in plants due to the lack of Boron, Zinc, Manganese, Iron, Copper, Molybdenum, Chlorine.

Micronutrients, are the building blocks for good health. People who do not have enough of these essential nutrients develop micronutrient malnutrition, which can be

devastating. So it is extremely important to detect the hidden hunger.

1.3 OBJECTIVES

- ✓ The main objective of our project is to detect hidden hunger at early stages using image processing techniques.
- ✓ To develop deep learning models for determining the micronutrient deficiency in human and plants.
- ✓ To analyze the performance of deep learning approaches for micronutrient deficiency detection.
- ✓ To create a web page for determining the micronutrient deficiency.

1.4 ORGANIZATION OF CHAPTERS

Literature survey is explained in chapter 2. In chapter 3, methodology used for the proposed model is described. Results and discussion is presented in chapter 4. In chapter 5, summary, conclusion and references are explained.

CHAPTER 2

LITERATURE REVIEW

2.1 GENERAL

In this chapter, a brief discussion about the various techniques used for the classification of micronutrient deficiency detection is made.

Mayuri Sharma et al., (2022) proposed a system based on an ensemble learning framework that employs transfer learning architectures to diagnose rice plant deficiencies. Six TL architectures viz. Xception, DenseNet201, InceptionResNetV2, InceptionV3, ResNet50V2, and VGG19 were employed to perform this task. The ensemble-based architecture enhanced the highest classification accuracy to 100% from 99.17% in the Mendeley dataset, while for the Kaggle dataset; it was enhanced to 92% from 90%.

Trung-Tin Tran et al., (2019) proposed to prevent pathologies caused by the lack of nutrients, designed the two forecast models based on the CNN structure which are capable models for predicting nutrient deficiencies in tomato plants. They suggested using ensemble learning to enhance the accuracy of the results. Moreover, based on these results, it is possible to apply the forecasting model in cultivating tomato plants under greenhouse conditions in order to minimize the negative impacts of nutrient deficiencies, contributing to an increase in harvest rate. The observation results, which were evaluated under real conditions, show that accuracy rates increased from 87.27% for Inception-ResNet v2 and 79.09% for Autoencoder to 91% for ensemble averaging.

Mohamed Farag Taha et al., (2022) aimed to integrate color imaging and deep convolutional neural networks (DCNNs) to diagnose the nutrient status of

lettuce grown in aquaponics. Their approach consists of multi-stage procedures, including plant object detection and classification of nutrient deficiency. The robustness and diagnostic capability of proposed approaches were evaluated using a total number of 3000 lettuce images that were classified into four nutritional classes namely, full nutrition (FN), nitrogen deficiency (N), phosphorous deficiency (P), and potassium deficiency (K). The segmentation model obtained an accuracy of 99.1%. Also, the deep proposed classification model achieved the highest accuracy of 96.5%.

Jinhui Yi et al., (2020) proposed a new dataset consisting of RGB images that were taken from sugar beets that received seven types of nutrient treatments. The dataset is a unique benchmark since the crop was growing as part of long-term fertilizer experiments under real field conditions. As baseline, they evaluated five convolutional neural networks that provide different trade-offs between accuracy and processing speed. While the best network was able to recognize the symptoms of nutrient deficiencies in sugar beets with a high accuracy if all development stages have been observed during training, recognizing nutrient deficiencies across crop development stages remains an open problem.

Hamna Waheed et al., (2022) presented a deep artificial neural network and deep learning-based methods for the early detection of diseases, pest patterns, and nutritional deficiencies. They used a real-field dataset consisting of healthy and affected ginger plant leaves. The results show that the convolutional neural network (CNN) has achieved the highest accuracy of 99% for disease rhizomes detection. For pest pattern leaves, VGG-16 models showed the highest accuracy of 96%. For nutritional deficiency-affected leaves, ANN has achieved the highest accuracy (96%). In this study, a novel and unique dataset was developed from the orchard of Pir Mehr Ali Shah Arid Agriculture University Rawalpindi (PMAS-AAUR). The study applied ANN, CNN, MobileNetV2, and VGG-16 to classify the images in the

dataset. These algorithms were used to detect pest patterns, deficiency nutrients, and soft rot disease of ginger.

Sourabhi Debnath et al., (2021) proposed a vineyard management system whose efficiency is directly related to the effective management of nutritional disorders, which significantly downgrades vine growth, crop yield and wine quality. To detect nutritional disorders, they successfully extracted a wide range of features using hyperspectral (HS) images to identify healthy and individual nutrient deficiencies of grapevine leaves. Features such as mean reflectance, mean first derivative reflectance, variation index, mean spectral ratio, normalized difference vegetation index and standard deviation were employed at various stages in the ultraviolet, visible and near-infrared regions for the experiment. Leaves were examined visually in the laboratory and grouped as either healthy or unhealthy. Then, the features of the leaves were extracted from these two groups. In a second experiment, features of individual nutrient-deficient leaves (e.g., N, K and Mg) were also analyzed and compared with those of control leaves. Furthermore, a customized support vector machine was used to demonstrate that these features can be utilized with a high degree of effectiveness to identify unhealthy samples and not only to distinguish from control and nutrient deficient but also to identify individual nutrient defects.

Nurbaity Sabri et al., (2022) proposed a system to help agriculturist, farmers and researchers to identify the type of maize nutrient deficiency to determine an action to be taken. This research uses image processing techniques to determine the type of nutrient deficiency that occurs on the plant leaf. A combination of Gray-Level Co-Occurrence Matrix (GLCM), hu-histogram and color histogram has been used as a parameter for further classification process. Random forest technique was used as classifiers manage to achieve 78.35% of accuracy. It shows random forest is a suitable classifier for nutrient deficiency detection in maize leaves. This research

classifies four type of class which is healthy leaf, nutrient, magnesium and potassium. Some of the nutrient deficiency is likely to have the same trait, therefore it is difficult to classify these nutrient.

Anu Jose et al., (2021) created an artificial neural network model to recognize and classify the nutrient deficiency in tomato by examining the leaf characteristics. This will help farmers to adjust the nutrient supply to the plant. If soil lacks a specific nutrient, it will reflect in the physical characteristics of a leaf. The color and shape of a leaf are the two major features used for identifying the nutrient deficiency. The comparison of different segmentation schemes like hue based and threshold based schemes shows their influence in the performance of the proposed system. The influence of different activation functions in the artificial neural network is also studied in this work. The proposed system is an image processing model consisting of features from hue is the best model for the task of tomato leaf classification. Elimination of intensity in texture feature calculation is the major advantage. It nullifies the effect of lighting variations in an outdoor environment. The comparison of different segmentation schemes like hue based and threshold based schemes shows their influence in the performance of the proposed system. Artificial neural network has given an accuracy of 88.27% for detecting and classifying nutrient deficiencies.

Lili Ayu Wulandhari et al., (2019) proposed a deep convolutional neural network to diagnose condition based on image of plants. Inception-Resnet architecture with transfer learning and fine tuning from model that is previously trained using ImageNet dataset is applied in the experiments. They trained the model further with image of okra plants. Image augmentation method is also used to increase the variation of the dataset. The experimental results show that fine tuning approach achieves the best accuracy, namely 96% and 86% for training and testing respectively. Improvement of this work is conducted by applying simpler

architecture such as Mobile net to be easier implemented in Mobile platform.[9]

Choi Jae-Won, Tin Tran Trung et al., (2018) proposed a method of recognition and prediction the nutrient deficiency occurs on fruiting phrase of tomato plant based on the deep neural network. They used two essential mineral nutrients (i.e. Calcium and Potassium) for evaluating the nutrient status in the development of tomato plant. Inception-ResNet v2 based-Convolution Neural Network (CNN) is applied to distinguish the above mineral nutrients with the captured images of tomato plant growth under the greenhouse conditions. The authors focused on analyzing and extracting the captured image of tomato fruits in its growth process through the nutrient status. In this state-of-the-art technology, the proposed approach reaches the high accuracy as much as possible. Moreover, the proposed system can adapt to upgrade the nutrient status monitoring system on the large scale of the greenhouse. As future work, the proposed system with AI algorithm that provides the ability to develop the monitoring system on controlling the lack of nutrition during the tomato growth process.

Yu-Jiun Fan et al.,(2023) proposed a system based on an machine learning framework that employs a deep convolutional neural network architecture for Pectus Excavatum screening through frontal chest radiography, which is the most common imaging test in current hospital practice. Convolutional neural network architecture Xception were employed to perform this task. Among them, 80% were used as the training set used to train the established CNN algorithm, Xception, whereas the remaining 20% were a test set for model performance evaluation. The test accuracy of the model reached 0.989, and the sensitivity and specificity were 96.66 and 96.64, respectively. The highest accuracy for the model, evaluated on the test set, was realized after 28 epochs on the training set. The test set used for evaluating the performance of our model was not used previously during the training process. In addition to the Chest X-ray, there are several large open-access chest X-

ray datasets worldwide that may be utilized for future work, such as CheXpert from Stanford , MIMIC-CXR from M.I.T and the well-known Alicante hospital chest X-ray datasets with large chest X-ray image data.

Wannipa Sae-Lim et al., (2019) Proposed a modified Mobile Net which is CNN architecture for skin lesion classification. This architecture was improved from a classical MobileNet by modifying the layer of MobileNet and also reduced the total number of parameters that made this approach run faster. MobileNet uses depthwise separable convolutions initially introduced into decrease the computational task in the first few layers. In this work, Human Against Machine with 10,000 training images (HAM10000) dataset was used. The dataset consists of 10,015 dermatoscopic images which contain seven diagnosis of pigmented skin lesions. The original resolution of image was 450x600 pixels resized to 244x244, in three channels of RGB colour compatible with input layer of the models. There are two experiments done in this model one is Dataset without any pre-processing technique was used to train and evaluate MobileNet and the modified MobileNet. The modified model achieves the accuracy rates of 83.93% in Experiment–1 and 83.23% in Experiment–2, which are 3.79% and 3.89% higher than MobileNet, respectively. In addition, this modified model uses fewer parameters than classical MobileNet. Thus, the computational time was reduced.

J. Banumathi et al.,(2021) proposed a Xcep-HSI model performs the FRML map extraction process and then Xception based feature extraction task takes place to derive a set of feature vectors. At last, the QPSO-KELM model is applied to identify the different class labels of HSI. The processes involved in FRML map extraction are given as follows: Initially, the pixels in the image are stored as a high dimensional vector where each element signify the spectral features in every band. Then, the values of every two elements are determined using a normalized difference index (NDI) for constructing a 2-D matrix known as feature

relation matrix, subsequently, a matrix is converted into an image for building FRM. The performance of the Xcep-HSIC model is simulated using Intel i5, 8th generation PC with 16 GB RAM, MSI L370 Apro, Nvidia 1050 Ti4 GB. For simulation, Python 3.6.5 tool is used along with pandas, sklearn, Keras, Matplotlib, TensorFlow, opencv, Pillow, seaborn and pycm. For experimentation, 50% of data is used for training and testing correspondingly. The Xcep-HSIC model has reached a higher validation accuracy of 0.9432. The loss graph tends to be decreased with an increase in epoch count, the training and validation accuracies are enhanced with maximization in the count of epochs. The Xcep-HSIC technique accomplishes a minimum validation loss of 0.275. This paper has developed a new model for HSI analysis and classification using the XcepHSIC model. The simulation outcome verified the supremacy of the Xcep-HSIC technique over the existing methods with the maximum accuracy of 94.32% and 92.67% on the applied India Pines and Pavia University dataset.

CHAPTER 3

METHODOLOGY

3.1 GENERAL

Convolution Neural Network (CNN) is one of deep learning methods which is used to obtain spatial information in an image. CNN has several architectures which were introduced by previous researchers to reach better performance in image recognition. This research applies Resnet152V2, InceptionV3, Xception, MobileNet, CNN model as the algorithms for detecting micro nutrient deficiency based on human nail, eye and plant images.

3.2 BLOCK DIAGRAM

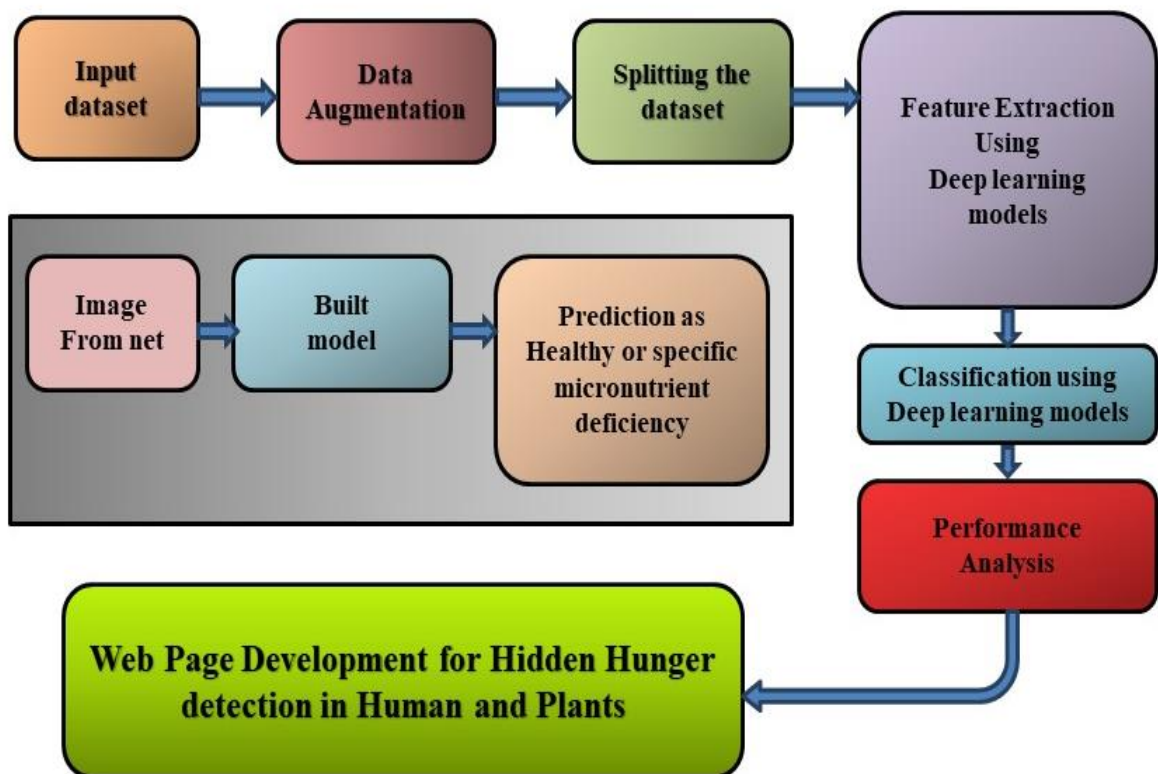


Figure 3.1: Block Diagram of the Hidden Hunger detection system.

3.3 INPUT DATASET

Micro nutrient deficiency can be identified in plants and human with visible symptoms. In human specific micronutrient deficiency can be identified with visible symptoms appearing on nails, tongue, skin, eye etc. Nail disease detection dataset is obtained from Roboflow. Here we are using images of various nail signs like white spots, vertical and horizontal lines on nails, Brittle nails etc. in which each nail pattern indicates different micro nutrient deficiency like Zinc, Vitamin D, A, Iodine etc. Similarly we use eye disease detection dataset obtained from Kaggle that has classes of various eye disease like glaucoma, cataract etc. from which we chose particular class called Uveitis. People who are Vitamin D deficient suffer from Uveitis. So uveitis images are kept under class Vitamin D. For Plants the input dataset of micronutrient deficiency is obtained for the banana leaves. There are five different classes in the leaf dataset like boron, iron, manganese, zinc, healthy. Leaf dataset is obtained from Mendeley.

3.4 DATA AUGMENTATION

Data augmentation is a technique of artificially increasing the training set by creating modified copies of a dataset using existing data. It includes making minor changes to the dataset or using deep learning to generate new data points. Data augmentation tool is an open-source tools that we can use to perform various data augmentation techniques and improve the model performance. One such tool we use is Augmentor. It is a Python package for image augmentation and artificial image generation. We can perform Perspective Skewing, Elastic Distortions, Rotating, Shearing, Cropping, and Mirroring. Augmentor also comes with basic image pre-processing functionality. The augmentation may be of geometric transformation and color space transformation. Geometric transformation include flip, crop, rotate etc. and color space transformation includes change the brightness, RGB color channel and contrast of an image

3.4.1 Image Augmentation Techniques

Augmentation techniques that is used in this project are

- ✓ Zoom
- ✓ Flip
- ✓ Random Brightness
- ✓ Random Distortion

❖ **Zoom** - The zoom augmentation method is used to zooming the image. This method randomly zooms the image either by zooming in or it adds some pixels around the image to enlarge the image.

❖ **Flip** - Flipping means rotating an image in a horizontal or vertical axis. In this project we flipped from top-bottom. The Vertical flip is a data augmentation technique that takes both rows and columns of such a matrix and flips them vertically. As a result we obtain an image flipped top-bottom along the x-axis.

❖ **Random Brightness** - Random brightness image augmentation is used to generate images with varied brightness levels for feeding our deep learning models. In this method we will train the system with different colour intensity images, so that the model can generalize the images which are not trained even on different lighting conditions. So this method can make the image a little darker, brighter, or both.

❖ **Random Distortion** - Distortion is when the straight lines of an image appear to be deformed or curved unnaturally.

3.5 SPLITTING THE DATASET

Data splitting is dividing dataset into two or more subsets. Typically, with a two-part split, one part is used to evaluate the model and the other to train the model. Data should be split such that data sets have a high amount of training data. Generally it can be split in the ratio of 7:2:1 for train, test and validation. Here we separate the dataset as train, test and validation in the ratio of 7:2:1 (train : validation

: test) for leaf dataset and 8:2 (train : validation) for nail dataset.

3.6 FEATURE EXTRACTION USING DEEP LEARNING APPROACHES

- Feature extraction is a part of the dimensionality reduction process, in which, an initial set of the raw data is divided and reduced to more manageable groups. So when we want to process it will be easier. The most important characteristic of these large data sets is that they have a large number of variables. These variables require a lot of computing resources to process. So Feature extraction helps to get the best feature from those big data sets by selecting and combining variables into features, thus, effectively reducing the amount of data. These features are easy to process, but still able to describe the actual data set with accuracy and originality. The layers of a deep learning model that are involved in feature extraction are explained below.

➤ **Input layer:** This layer takes the input image and prepares it for processing by the neural network. In most of the deep learning used, the input layer takes an RGB image with a resolution of 224x224 pixels.

➤ **Convolutional layers:** These layers perform convolutions on the input image to extract features. Convolution may be performed by using a combination of 3x3 and 1x1 convolutional layers, with stride and padding which is used to control the size of the feature maps.

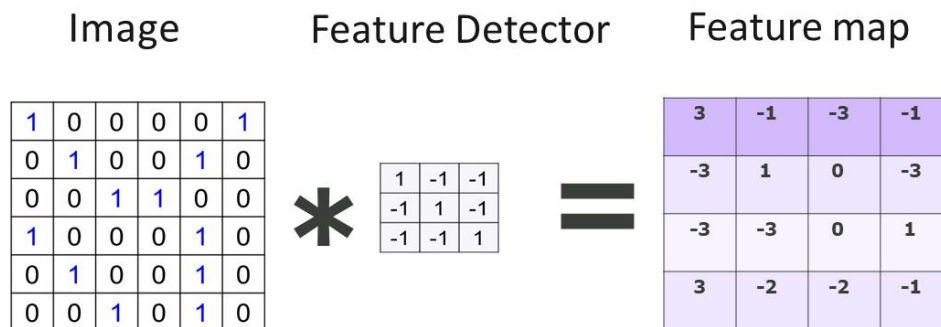


Figure 3.2: Convolution operation.

➤ **Bottleneck residual blocks:** These blocks are the main building blocks

of the ResNet-152 v2 architecture. They consist of a sequence of 1x1, 3x3, and 1x1 convolutions, with batch normalization and ReLU activation functions applied between each convolution. The bottleneck structure reduces the computational cost of the model and improves its accuracy.

- **Pooling:** The purpose of the pooling layers is to reduce the dimensions of the hidden layer by combining the outputs of neuron clusters at the previous layer into a single neuron in the next layer.

- **Max pooling layers:** These layers down sample the feature maps by taking the maximum value in each pooling region. Generally, max pooling with a kernel size of 3x3 and stride of 2 is used.

- **Global average pooling layer:** This layer performs average pooling across the entire feature map to obtain a single feature vector for the entire image.

- **Fully connected layer:** These layers take the output of the previous layers and perform a linear transformation to produce the final output logics. As a result, all possible connections layer-to-layer are present, meanwhile every input of the input vector influences every output of the output vector.

- **Relu:** The ReLU function is another non-linear activation function that has gained popularity in the deep learning domain. ReLU stands for Rectified Linear Unit. The main advantage of using the ReLU function over other activation functions is that it does not activate all the neurons at the same time.

This means that the neurons will only be deactivated if the output of the linear transformation is less than 0.

$$f(x) = \max(0, x) \dots\dots\dots(3.1)$$

Where,
 x is the input to the activation function.
 $f(x)$ is the ReLU function.

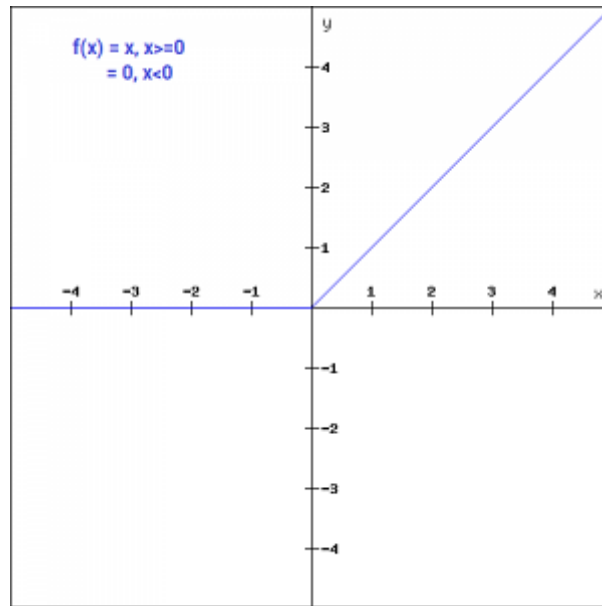


Figure 3.3: RELU graph

The above figure shows that if input to the activation function less than zero then the output will be zero, and if the input is greater than or equal to zero then the function returns the same input as output. This ReLU helps to overcome the vanishing gradient problem.

3.6.1 Resnet

Residual Network (ResNet) is a deep learning model used for computer vision applications. It is a Convolution Neural Network (CNN) architecture designed to support hundreds or thousands of convolution layers. Previous CNN architectures were not able to scale to a large number of layers, which resulted in limited performance. However, when adding more layers, researchers faced the “vanishing gradient” problem. ResNet provides an innovative solution to the vanishing gradient problem, known as “skip connections”. ResNet stacks multiple identity mappings (convolution layers that do nothing at first), skips those layers, and reuses the activations of the previous layer. Skipping speeds up initial training by compressing the network into fewer layers. Then, when the network is retrained, all layers are expanded and the remaining parts of the network—known as the residual parts—are

allowed to explore more of the feature space of the input image.

- **Residual Blocks**

Residual blocks are an important part of the ResNet architecture. It works with a small number of convolutional layers. However, subsequent research discovered that increasing the number of layers could significantly improve CNN performance. The ResNet architecture introduces the simple concept of adding an intermediate input to the output of a series of convolution blocks.

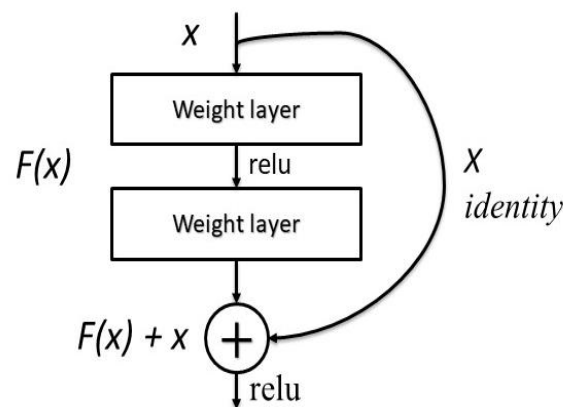


Figure 3.4: Residual Block

The image above shows a typical residual block. This can be expressed in Python code using the expression,

$$\text{output} = F(x) + x \quad \dots\dots\dots (3.2)$$

where x is an input to the residual block and output from the previous layer, and $F(x)$ is part of a CNN consisting of several convolution blocks. This technique smooths out the gradient flow during back propagation, enabling the network to scale to 50, 100, or even 150 layers. Skipping a connection does not add additional computational load to the network. This technique of adding the input of the previous layer to the output of a subsequent layer is now very popular, and has been applied to many other neural network architectures including UNet and Recurrent Neural Networks (RNN).

- **RESNET 152V2**

Residual Neural Network (ResNet152V2) is a convolutional neural network which has 152 layers in it. To solve the problem of vanishing gradients ResNet uses skip connection for fitting the output from previous layer to next layer. It has been the winner ImageNet challenge in 2015. It is used to compare the accuracy of the proposed model. It is a pretrained model that has several convolution layers and max pooling layers. Arguments of the ResNet function are include top, weights, input tensor, input shape, pooling, classes, classifier activation. Resnet152v2 model input image is processed by the 152 convolutional layers and image is reshaped. Then the image is sent to the hidden layer for further processing and the output is obtained from the output layer.

•RESNET Architecture

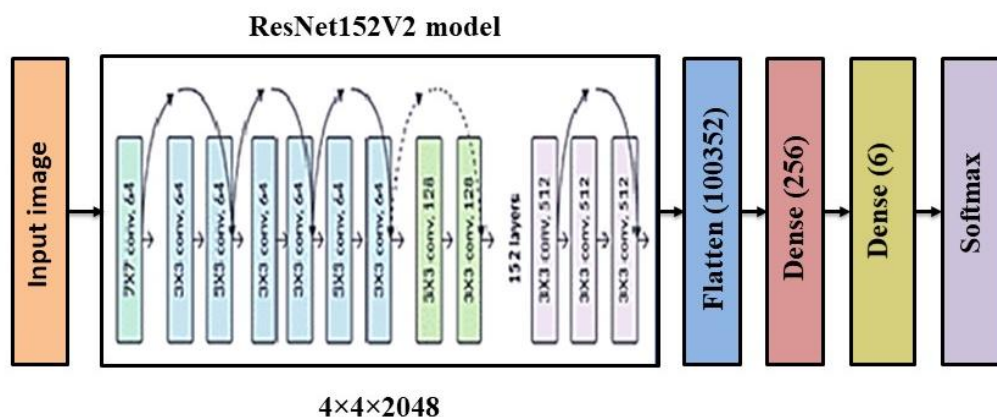


Figure 3.5: ResNet Architecture

The Figure 3.5 shows the architecture of ResNet152V2. Here the first layer is the input layer where an image of dimension 224x224x3 is fed and followed by that layer there are 152 convolution layers. There is a fully connected layer and a dense layer where softmax activation function is used to get the output of image classification problem.

3.6.2 InceptionV3

The inception v3 model was released in the year 2015, it has a total of 42 layers and a lower error rate than its predecessors. Let's look at what are the different optimizations that make the inception V3 model better. The inception V3 is a superior version of the basic model InceptionV1 which was introduced as GoogLeNet in 2014. As the name suggests it was developed by a team at Google. Inception-v3 is a convolutional neural network architecture from the Inception family that makes several improvements including using Label Smoothing, factorized 7 x 7 convolutions, and the use of an auxiliary classifier to propagate label information lower down the network

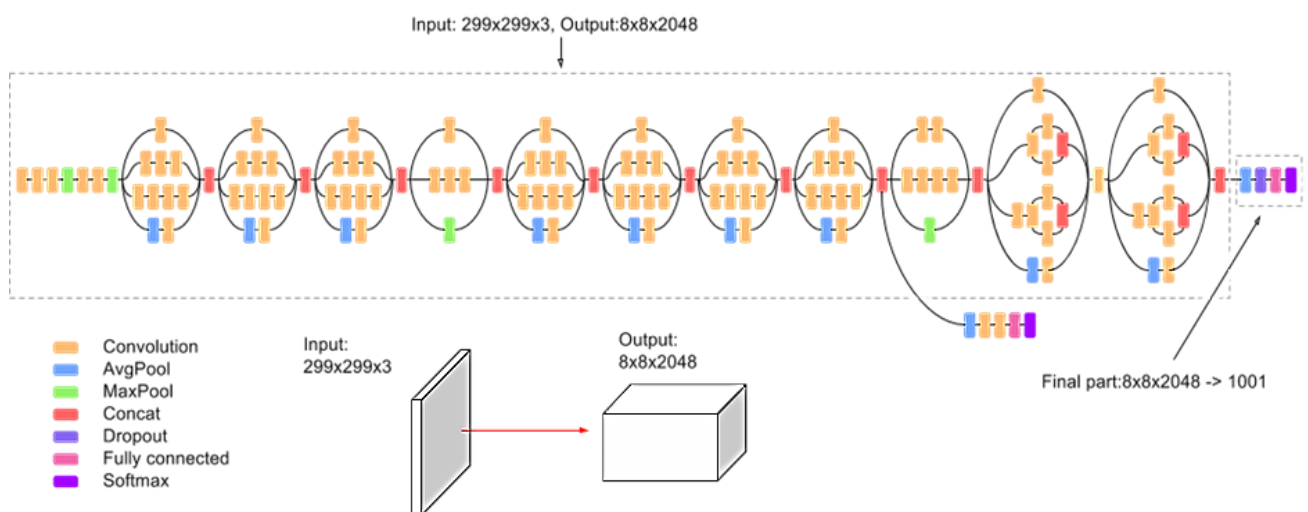


Figure 3.6: Inception V3 Architecture

3.6.3 Xception

Xception stands for “extreme inception”, it takes the principles of inception to an extreme. In Inception, 1x1 convolutions were used to compress the original input, and from each of those input spaces we used different type of filters on each of the depth space. Xception just reverses this step. Instead, it first applies the filters on each of the depth map and then finally compresses the input space using 1X1 convolution by applying it across the depth. This method is almost identical to a

depth wise separable convolution. There is one more difference between Inception and Xception. The presence or absence of a non-linearity after the first operation. In Inception model, both operations are followed by a ReLU non-linearity, however Xception does not introduce any non-linearity.

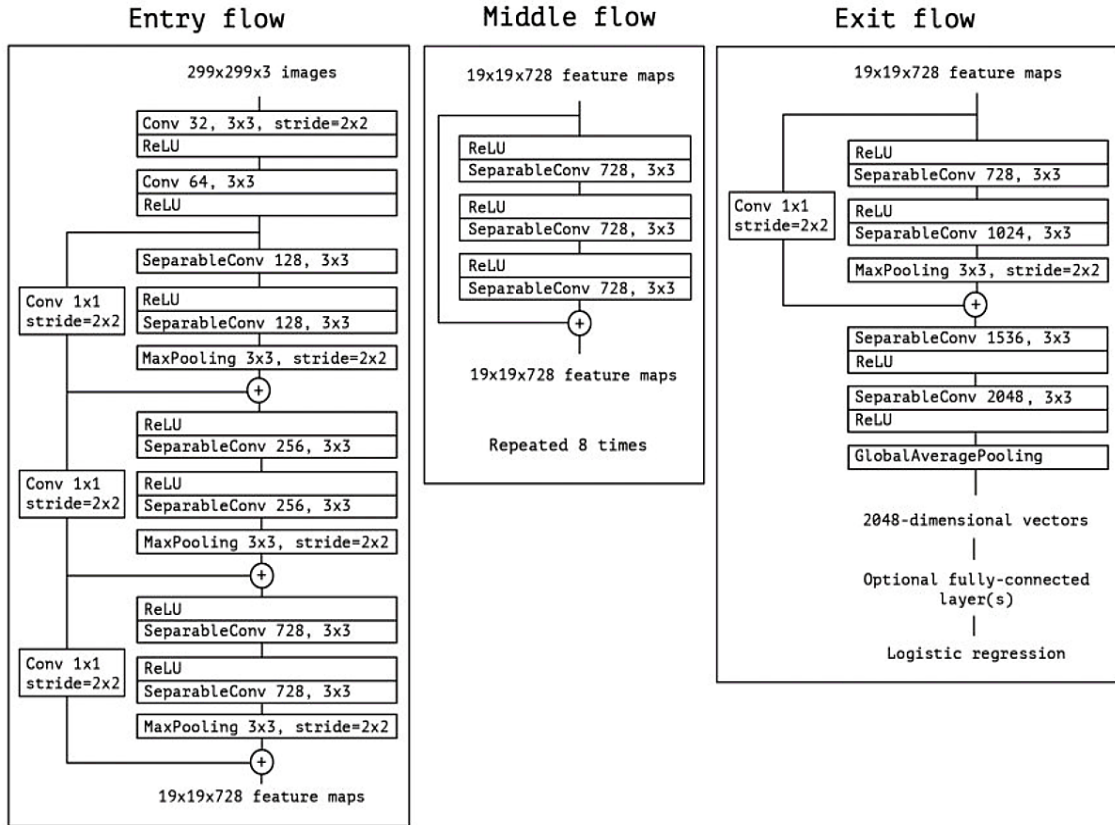


Figure 3.7: Xception Architecture

From the above figure 3.6 the data first goes through the entry flow, then through the middle flow which is repeated eight times, and finally through the exit flow. Note that all convolution and separable convolution layers are followed by batch normalization. Xception architecture has overperformed VGG-16, ResNet and Inception V3 in most classical classification challenges.

3.6.4 MobileNetV2

MobileNet V2 is a type of convolutional neural network (CNN) architecture for mobile and embedded vision applications. MobileNet V2 is designed to be lightweight and efficient, making it ideal for applications on mobile devices and other resource-constrained platforms. It achieves this by using a combination of depthwise separable convolutions and linear bottlenecks. Depthwise separable convolutions are a type of convolutional operation that factorizes a standard convolution into two separate operations: a depthwise convolution that applies a single filter to each input channel, followed by a pointwise convolution that combines the outputs of the depthwise convolution. This allows for a significant reduction in computation and parameter size compared to standard convolutions. Linear bottlenecks are another technique used in MobileNet V2 to reduce the computational cost of the network. They involve using 1x1 convolution followed by a ReLU activation function to reduce the dimensionality of the input before applying a more computationally expensive operation such as a depthwise separable convolution. Overall, the combination of depthwise separable convolutions and linear bottlenecks in MobileNet V2 results in a highly efficient and effective CNN architecture that can achieve state-of-the-art performance on a range of vision tasks while running quickly and using minimal computational resources. MobileNetV2 has 53 layers. It has convolution layer, Depth wise separable convolution layers where batch normalization and ReLu are used, global average pooling layer, and an output layer with softmax activation function. In convolution 2D ds layers, the number of filters are no need to be mentioned, instead it is applied separately to each filter. Batch normalization involves taking mean and variance of input data of all samples and this batch mean is divided by the standard deviation value. It can improve the stability and performance of the neural networks.

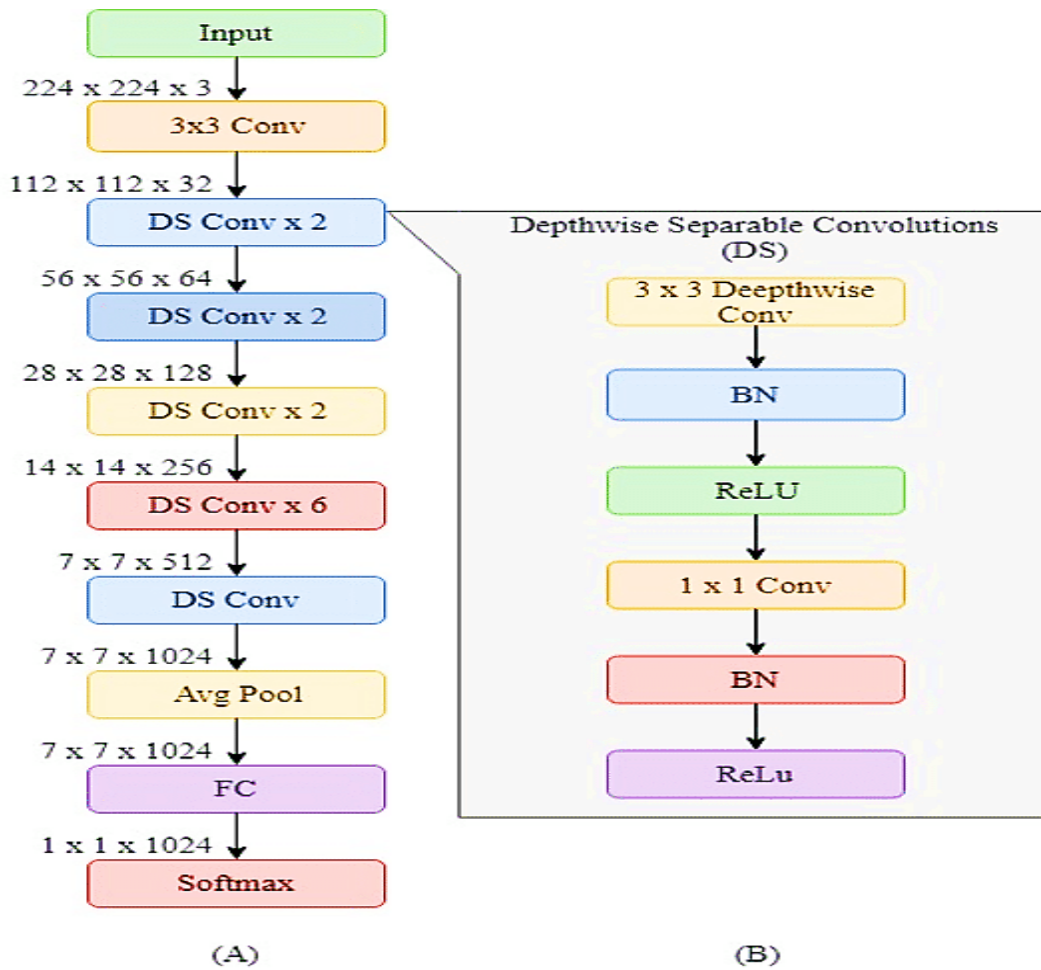


Figure 3.8: MobileNet V2 Architecture (A) MobileNet architecture, (B) Depth wise Separable layer.

In the above figure, (A) shows the architecture of MobileNetV2 which has convolution layers, depth wise separable convolution layers, average pooling layers, fully connected layer and an output layer with softmax activation function. The dimension of input image here is $224 \times 224 \times 3$ and after each convolution block the image size is reduced by half and before the fully connected layer the image size becomes $7 \times 7 \times 1024$, where 1024 is the number of filters used or number of output channels of that layer.

➤ MobileNet V2 model has 53 convolution layers and 1 Average Pool with nearly 350 GFLOP. It has two main components:

- Inverted Residual Block
- Bottleneck Residual Block

There are two types of Convolution layers in MobileNet V2 architecture:

- 1x1 Convolution
- 3x3 Depth wise Convolution

3.6.5 CNN model

We built a CNN model for the prediction of micronutrient deficiency in plants. The developed model has 5 convolution layers and 5 Max pooling layers. The convolution layers has filters of kernel size 3x3 with ReLu activation function being used. Max pooling layers uses pool size 2x2 and stride=2.

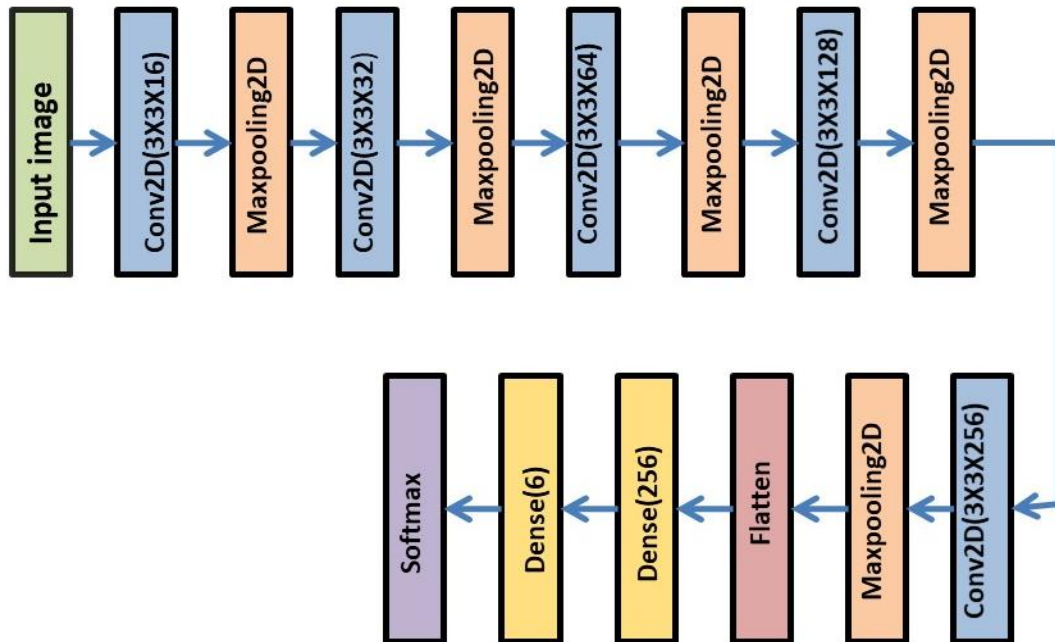


Figure 3.9: Architecture of CNN model

The dimension of input image is 224x224x3. This image is passed through several convolution and maxpooling layers to produce feature map. In this model, the number of filters in each convolution layer is gradually increased from 16,32,etc. and at the last convolution layer 256 filters are used. Next to this a flatten layer and a fully connected layer of 256 neurons were used and finally the output is classified using softmax activation function.

3.7 CLASSIFICATION USING DEEP LEARNING APPROCHES

- Classification is a supervised machine learning method where the model tries to predict the correct label of a given input data. In classification, the model is fully trained using the training data, and then it is evaluated on test data before being used to perform prediction on new unseen data.

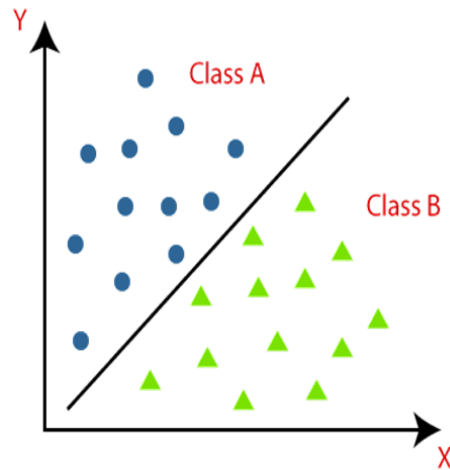


Figure 3.10: Basic classification approach

The above figure 3.10 shows that class A and class B are the two different classes having data points with similar properties and the classes are separated by a definite boundary. In deep learning models, classification can be performed by adding a dense layer with number of neurons equal to number of classes. This dense layer is the last output layer with activation function like softmax being used.

➤ **Activation function - Softmax:** The softmax function is a function that turns a vector of K real values into a vector of K real values that sum to 1. The input values can be positive, negative, zero, or greater than one, but the softmax transforms them into values between 0 and 1, so that they can be interpreted as probabilities. If one of the inputs is small or negative, the softmax turns it into a small probability, and if an input is large, then it turns it into a large probability, but it will always remain between 0 and 1.

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^k e^{z_j}} \dots\dots\dots(3.3)$$

Where, K = number of classes used.

z is the input vector to the softmax function.

z_i is the i th element of input vector.

$\sum_{j=1}^k e^{z_j}$ is the normalization term.

Hence the model classifies the input image to one among the five classes like Iodine, Iron, Vitamin B12, Vitamin D, Zinc, healthy for the hidden hunger detection in human. For hidden hunger detection in plants, the given input image is classified to one among the five classes like boron, manganese, iron, zinc and healthy.

3.8 PERFORMANCE ANALYSIS

Performance measurements are used to determine the effectiveness of image processing technique in achieving expected results. They are the quantities that are used to compare the performances of different system. It is a systematic approach to collecting, analyzing and evaluating how “on track” a project is to achieve its desired outcomes, goals and objectives.

➤ **Confusion matrix-** Confusion matrix is a technique for summarizing the performance of a classification algorithm. It is a table that shows correspondence between the classification result and a reference image. Confusion matrix is a useful machine learning method that allows to measure recall, F1 Score, ROC. The figure below shows the confusion matrix which is the matrix between true class and predicted class. From this matrix, other performance metrics like precision, recall, f1 score can be determined. Here true positives denote true class is equal to the predicted class.

		True Class	
		Positive	Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN

Figure 3.11: Confusion matrix

True positive (TP) - Observation is predicted positive and is actually positive.

False positive (FP) - Observation is predicted positive and is actually negative.

True Negative (TN) - Observation is predicted negative and is actually negative.

False Negative (FN) - Observation is predicted negative and is actually positive.

➤ **Recall** - Recall tell us about how well the model identifies true positive.

Recall is defined as the number of true positives divided by the total number of elements that actually belongs to the positive class.

$$\text{Recall} = \frac{\text{TP}}{\text{FN} + \text{TP}} \dots\dots\dots(3.4)$$

➤ **F1 score** - It is the harmonic mean of precision and recall. It takes both false positive and false negative into account therefore it performs well on an imbalanced dataset. F1 Score gives the weightage to recall and precision. It can be calculated by the following formula,

$$F_1 \text{ score} = \frac{2[\text{Precision} \times \text{Recall}]}{\text{Precision} + \text{Recall}} \dots\dots\dots(3.5)$$

The higher the precision and recall the higher the F1-Score. F1-Score ranges between 0 to 1. The closer it is to 1, the better the model.

➤ **Precision** – Precision is one indicator of a machine learning model's performance – the quality of a positive prediction made by the model. Precision refers to the number of true positives divided by the total number of positive

predictions

$$\text{Precision} = \frac{TP}{TP+FP} \dots\dots\dots(3.6)$$

➤ **Training loss** – The training loss indicates how well the model is fitting the training data. That is to say, it assesses the error of the model on the training set where training set is a portion of a dataset used to initially train the model.

➤ **Validation loss** – Validation loss is a metric used to assess the performance of a deep learning model on the validation set. The validation set is a portion of the dataset set aside to validate the performance of the model. The validation loss is similar to the training loss and is calculated from a sum of the errors for each example in the validation set.

➤ **Loss function** – A loss function measures how far the model deviates from the correct prediction. The loss function used in our deep learning model is Categorical cross entropy. It is used when adjusting model weights during training. The aim is to minimize the loss. The smaller the loss, the better the model. A perfect model has a cross-entropy loss of 0. It typically serves multi-class and multi-label classifications. Cross-entropy loss measures the difference between a deep learning classification model's discovered and predicted probability distributions.

$$loss = - \sum_{j=1}^k y_j \log(\hat{y}_j) \dots\dots\dots(3.7)$$

Where,

k is the number of classes used.

y_j is the actual value.

\hat{y}_j is the neural network prediction.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 GENERAL

In this chapter, the qualitative and comparative analysis of different deep learning model is done. And also the result in each step of our project is briefly discussed. To train and evaluate deep learning models we used google colab platform. It is a data analysis and machine learning tool that is used to combine the executable python code, images, HTML and more into a single document stored in google drive. Google colab has three run time types like standard, high-RAM, and GPU. We used GPU to run the code. For GPU instance, following specifications are there,

- ✓ CPU: 2vCPU
- ✓ GPU: 1NVIDIA Teslak80 GPU with 16GB OF VRAM
- ✓ RAM: 25.5GB




4.2 HIDDEN HUNGER DETECTION IN HUMAN

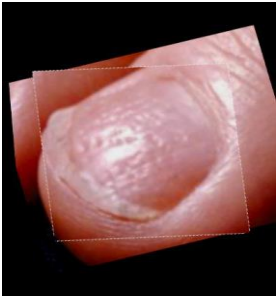
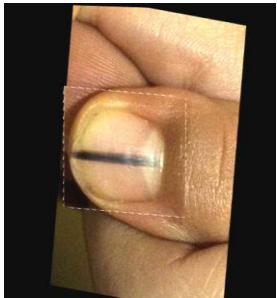


Micro nutrient deficiency can be identified from different visible changes in nails and eyes. Totally 3310 images were used for deficiency detection of after augmentation which 80% is used as training data, 20% is used as validation data. Dataset has 6 classes. Each class has images of specific micronutrient deficiency. The six Classes are healthy, Iodine, Iron, Vitamin B12, Vitamin D, Zinc. The dataset is taken from roboflow and Kaggle. The link for the dataset is given below.

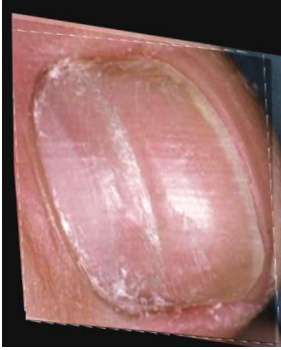
Roboflow - <https://universe.roboflow.com/knm/nail-disease-detection-mxoqy>

Kaggle-<https://www.kaggle.com/datasets/kondwani/eye-disease-dataset>

Table 4.1: Different classes in nail and eye dataset with their sample images.

SAMPLE IMAGE	CLASS NAME	VISIBLE SYMPTOM OF DEFICIENCY	NO OF IMAGES
	Healthy	—	558
			
	Iodine Deficiency	Clubbing—bulbous enlargement of the ends of one or more finger and toe nails.	574

	<p>Iron Deficiency</p>	<p>Pitting- ice pick-like depression in the nails.</p>	<p>567</p>
	<p>Vitamin D Deficiency</p>	<p>Melanoma- It can be identified from nails as a dark streak. may look like a brown or black band in the nail often on the thumb or big toe of dominant hand or foot. This dark streak can show up on any nail.</p>	<p>584</p>
		<p>Uveitis An inflammation of the middle layer of the eye (uvea). Symptoms include redness, pain, Light sensitivity, blurred vision, dark floating part in the field of Vision</p>	
	<p>Vitamin B-12 Deficiency</p>	<p>Blue Finger- It is characterized by an acute bluish discoloration of fingers which may be accompanied by pain. This is a case of a middle aged female who presented with painless bluish discoloration of</p>	<p>572</p>

		right hand and was diagnosed to have BFS.	
	Zinc Deficiency	Beau's lines are horizontal indentations, or ridges, that develop across the nails. They usually run straight across the nail. A person may develop one or more Beau's lines on any nail, or across multiple nails.	455

The above table shows the different classes of nail and eye images and its corresponding micro nutrient deficiency. This table also shows how many number of images are there in each class and also describes the visible deficiency symptoms in nail and eye. The images used for training are of dimension $224 \times 224 \times 3$.

4.2.1. Data augmentation

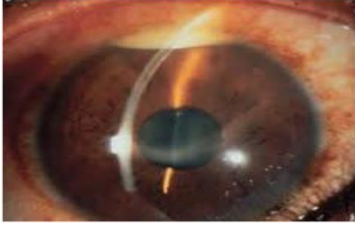

The first step of our project is to augment the dataset where the amount of dataset can be increased to desired level. Because of this augmentation the model could perform well. The following techniques are used for augmentation in the specified proportion, Zoom-20%, Flip-20%, Random distortion-30%, Random Brightness-10%. Initially, there are 3310 images and. It is augmented to 5000 images and the performance of the model is evaluated. Again, in order to improve the performance of model, the images are augmented to 7000 images and again performance of model is evaluated. It is inferred here that the model performed better with 7000 images than with 5000 images.

Table 4.2: Data Augmentation result

Classes	No of images before augmentation (3310)	No of images after Augmentation (5000)	No of images after Augmentation (7000)
Iodine	574	912	1189
Iron	567	851	1188
Vitamin B12	572	840	1236
Vitamin D	584	883	1194
Zinc	455	705	975
Healthy	558	809	1218

The above table shows that number of images before augmentation is 3310 and after augmentation it gets increased to 5000 images. When it is augmented for the second time it results with 7000 images. The table also shows number of images in each class before and after augmentation.

Table 4.3: Augmentation techniques

Augmentation Technique	Sample image	Probability of augmentation
Zoom		30%
Flip top-bottom		30%

Random Brightness		20%
----------------------	---	-----

The above table shows the different techniques used for augmentation. The sample images for each technique is also given in the table. This table also gives the probability of augmentation which means from the set of images 30% of them will be zoomed and 30% will be flipped and 20% will be subjected to brightness change.

4.2.2. Splitting the dataset

After augmentation data is split into train and test in the ratio of 8:2 where 80% is trained data and 20% is validation data. After augmenting dataset to 5000 images, there are 3998 images in train dataset and 1002 images in test dataset. When augmenting the same dataset to 7000 images, there are 5998 images in train dataset and 1002 images in test dataset. Splitting the dataset into 9:1 gave the best performance for ResNet152V2.

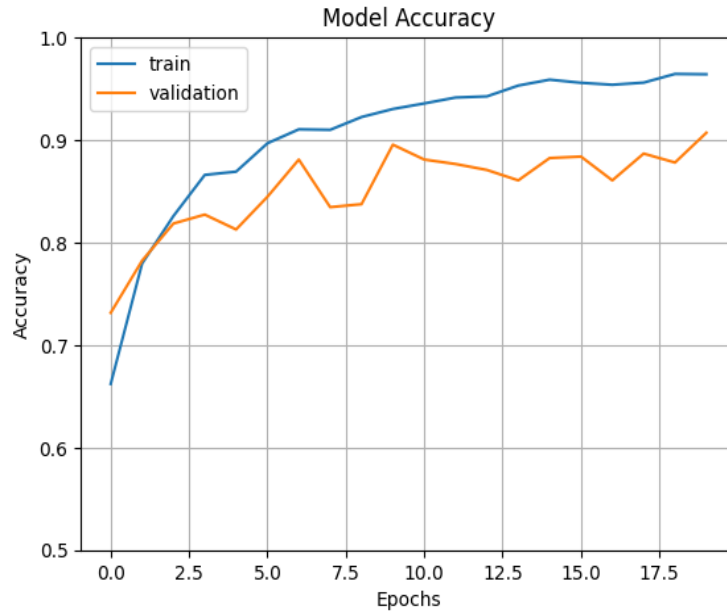
4.2.3. Deep learning models

We used four different models for feature extraction and classification.

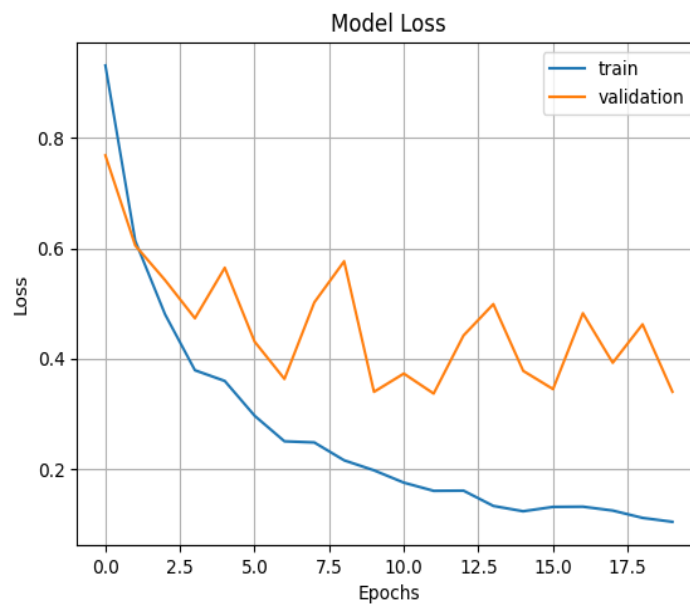
- ResNet 152 V2
- Inception V3
- Xception
- MobileNet V2
- Traditional CNN Method

➤ RESNET 152 V2

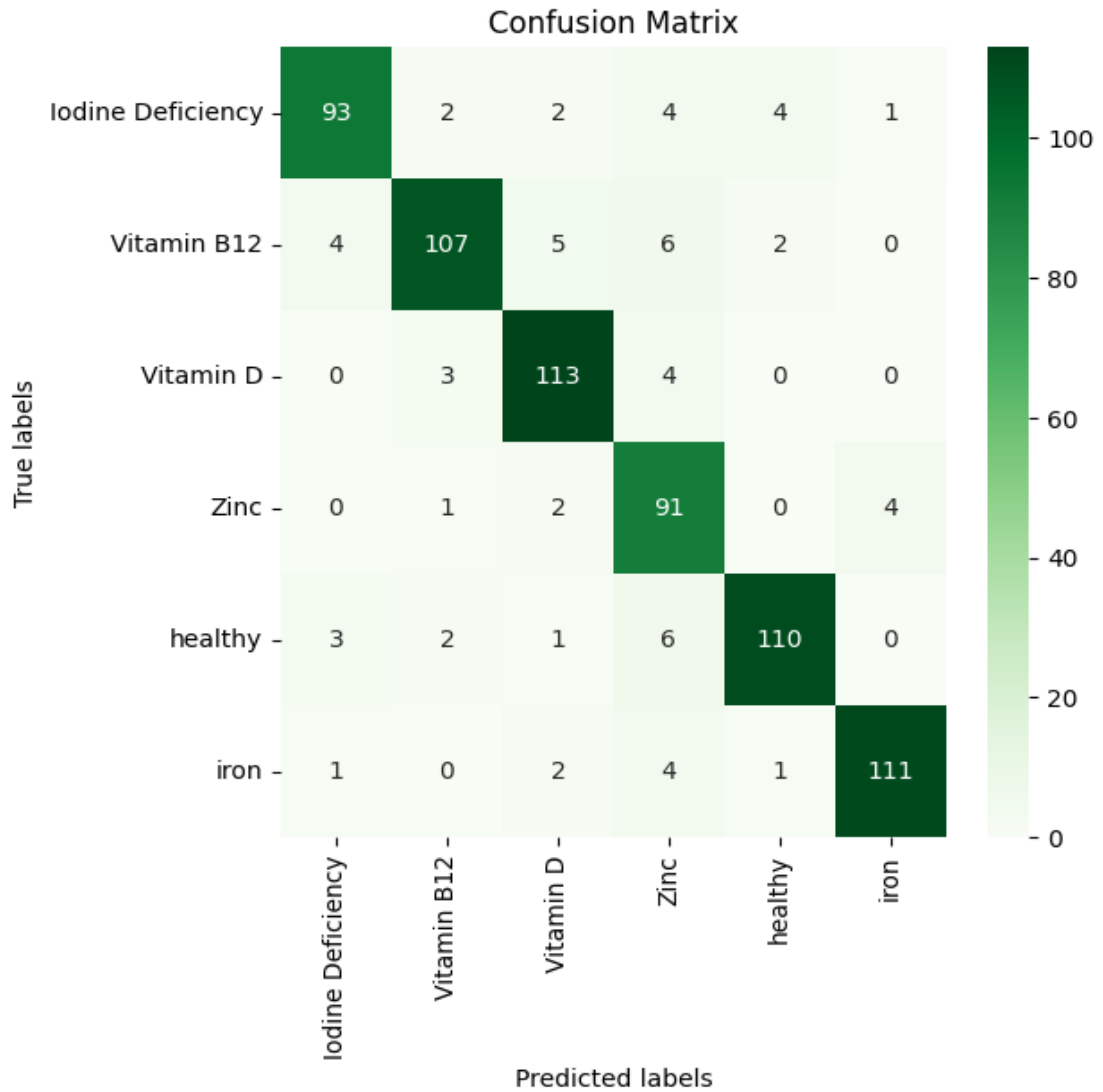
We use ResNet 152 V2 model for feature extraction and classification. This model gives the test accuracy of 90.71% and train accuracy is of 97.28%. The number of epochs used is 20, batch size is of 32, learning rate is 0.001 and Adam optimizer is used here.



(a) Accuracy plot



(b) Loss plot



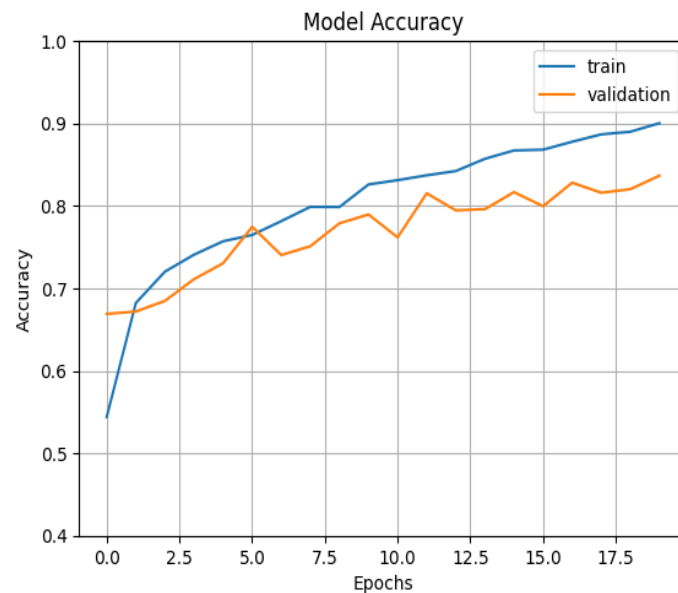
(c) Confusion matrix

Figure 4.1: Qualitative analysis of ResNET152V2

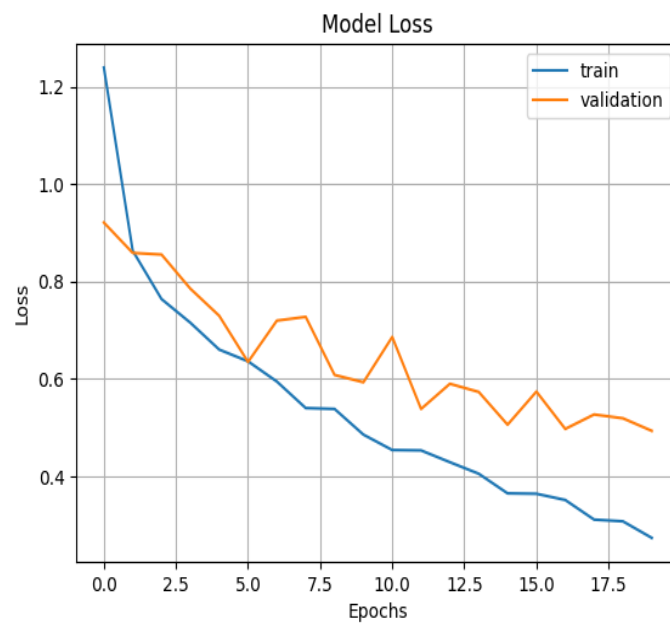
Figure 4.1 shows the accuracy plot where the validation accuracy and test accuracy increases as the number of epochs increases and the loss plot where the validation accuracy and test accuracy decreases as the number of epochs decreases. It also shows the confusion matrix which depicts how many images have been predicted correctly and those numbers are in the diagonal of matrix. Here the predicted label matches the true label. It is also inferred that few number of images of some classes are predicted as different class.

➤ INCEPTION V3

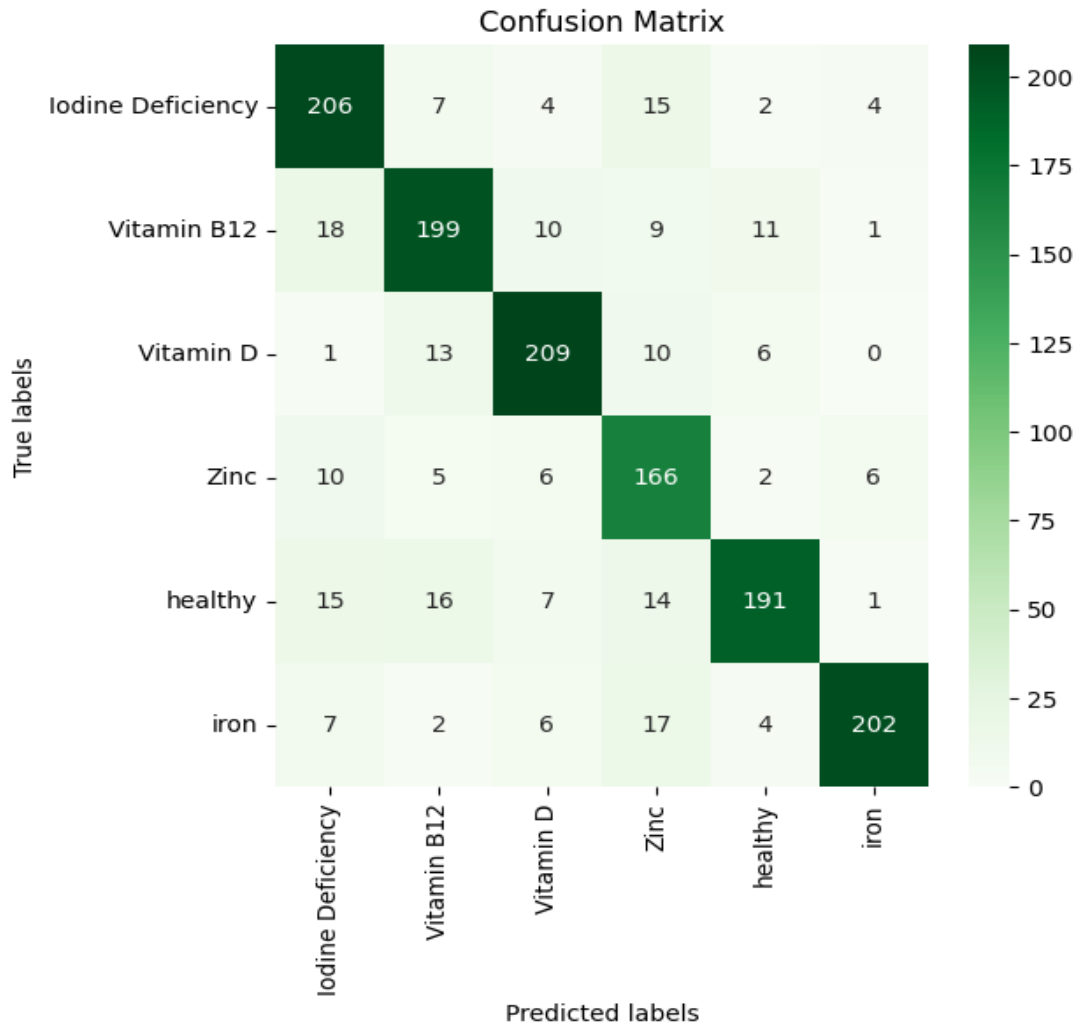
We use Inception V3 model for feature extraction and classification. This model gives the test accuracy of 83.67% and train accuracy is of 92.46%. The number of epochs used is 20, batch size is of 32, learning rate is 0.001 and Adam optimizer is used here.



(a) Accuracy Plot



(b) Loss Plot



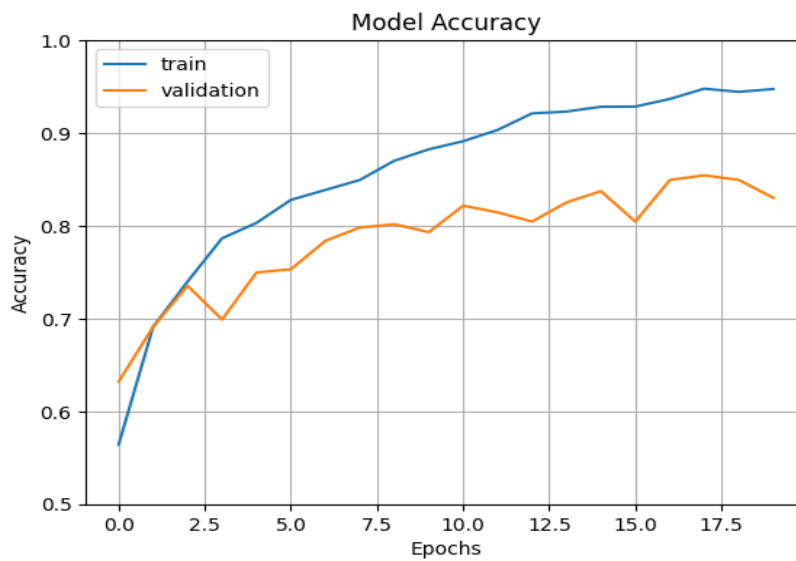
(c) Confusion matrix

Figure 4.2 Qualitative analysis of Inception V3

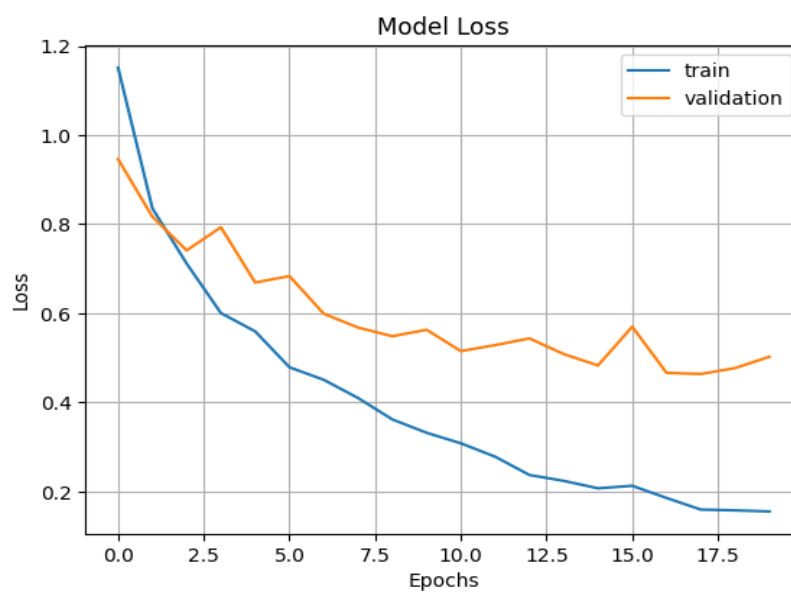
Figure 4.2 shows the accuracy plot where the validation accuracy and train accuracy increases as the number of epochs increases and the loss plot shows that the validation loss and train loss decreases as the number of epochs increases. It also shows the confusion matrix which depicts how many images have been predicted correctly and those numbers are in the diagonal of matrix. Here the predicted label matches the true label. It is also inferred that few number of images of some classes are predicted as different class.

➤ XCEPTION

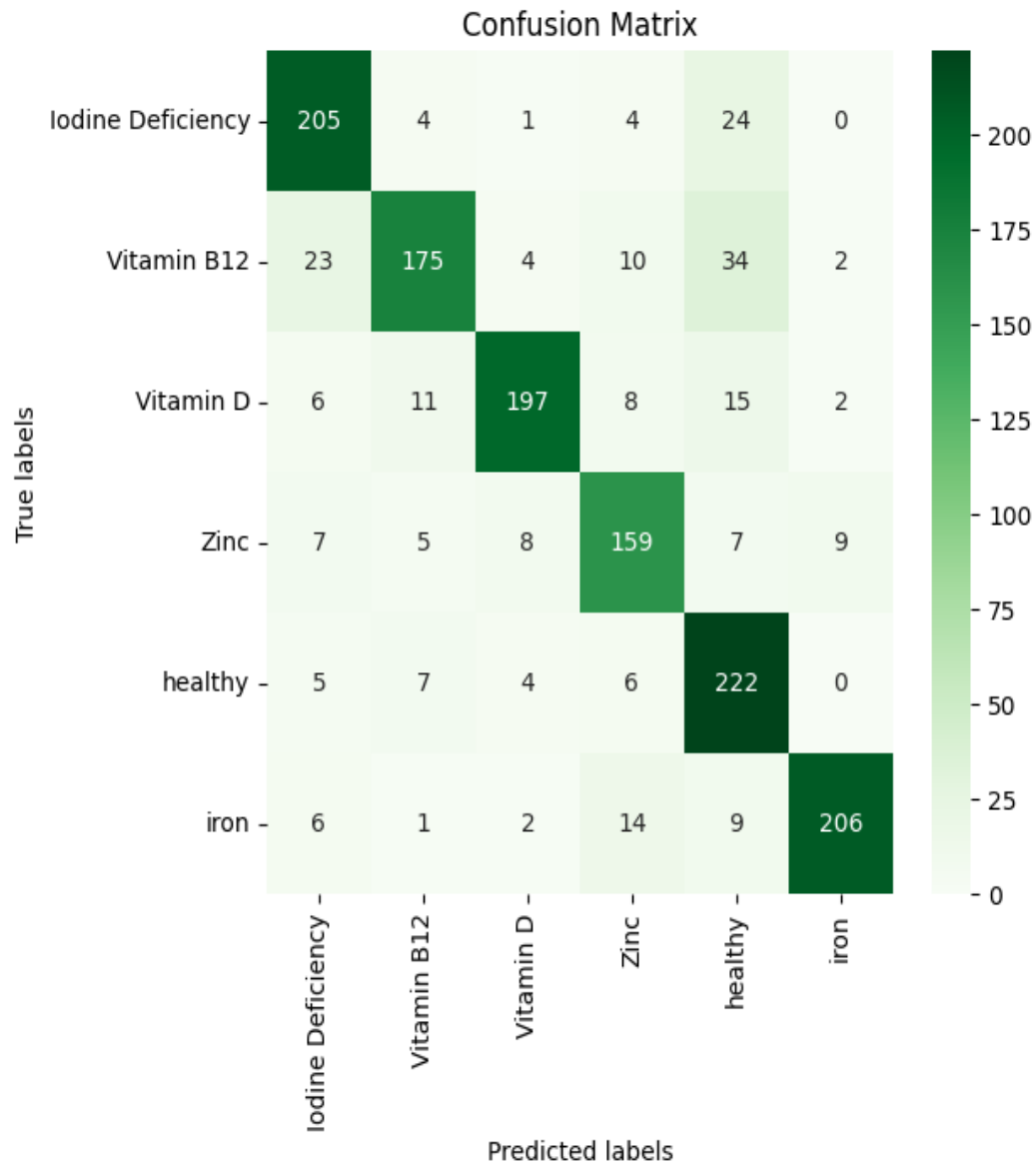
We use Xception model for feature extraction and classification. This model gives the test accuracy of 83.02% and train accuracy is of 94.52%. The number of epochs used is 20, batch size is of 32, learning rate is 0.001 and Adam optimizer is used here.



(a) Accuracy Plot



(b) Loss Plot



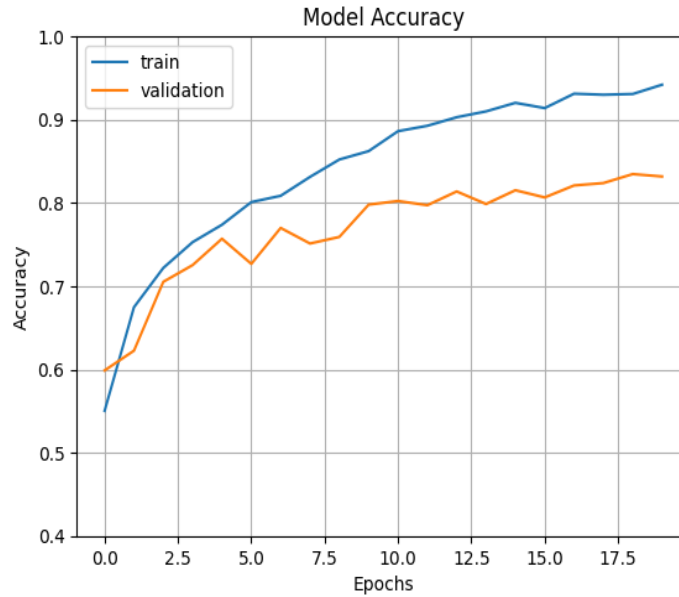
(c) Confusion matrix

Figure 4.3: Qualitative analysis of Xception

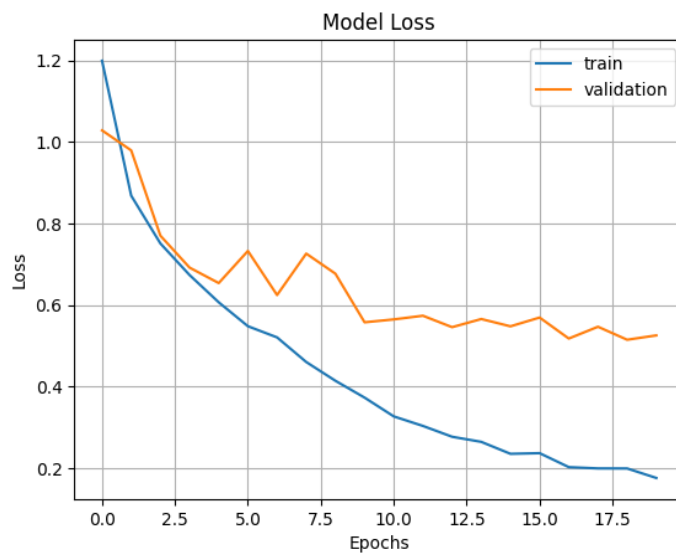
Figure 4.3 shows the accuracy plot where the validation accuracy and train accuracy increases as the number of epochs increases and the loss plot shows that the validation loss and train loss decreases as the number of epochs increases. It also shows the confusion matrix which depicts how many images have been predicted correctly and those numbers are in the diagonal of matrix. Here the predicted label matches the true label. It is also inferred that few number of images of some classes are predicted as different class.

➤ MOBILENET V2

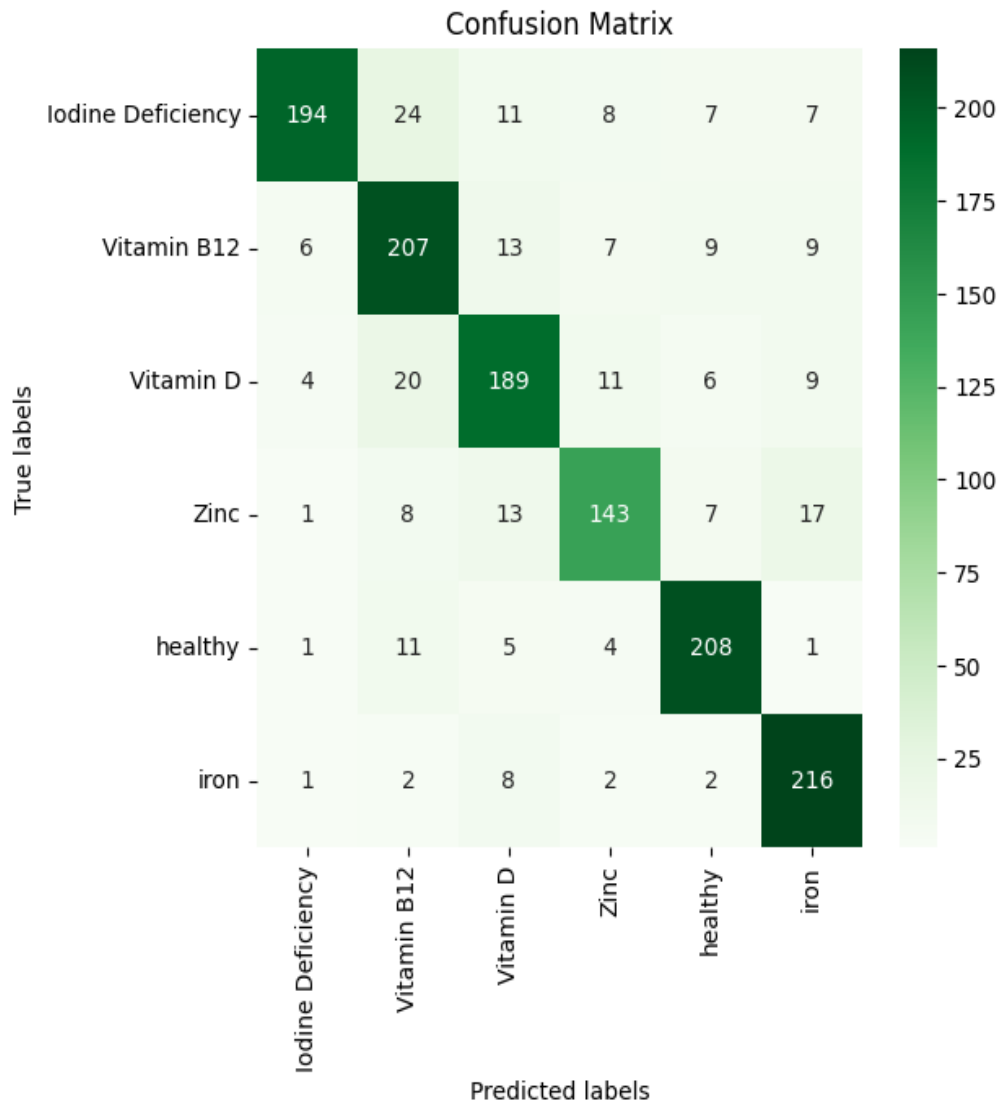
We use MobileNet V2 model for feature extraction and classification. This model gives the test accuracy of 83.18% and train accuracy is of 95.52%. The number of epochs used is 20, batch size is of 32, learning rate is 0.001 and Adam optimizer is used here.



(a) Accuracy plot



(b) Loss plot



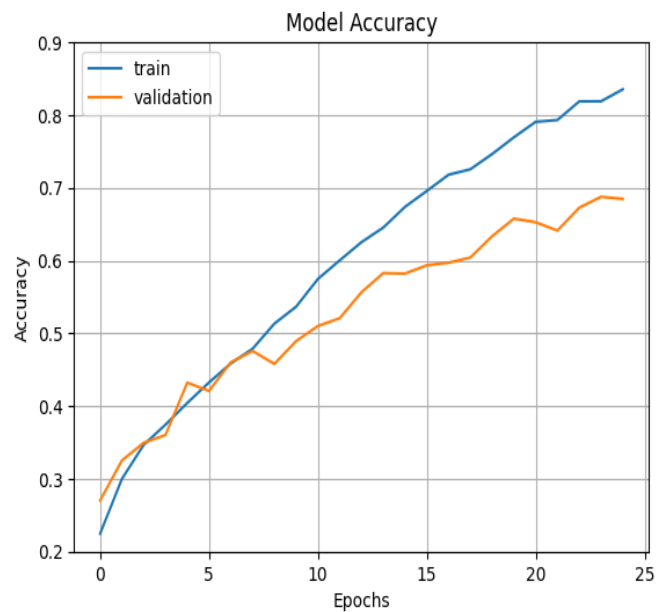
(a) Confusion matrix

Figure 4.4: Qualitative analysis of MobileNet V2

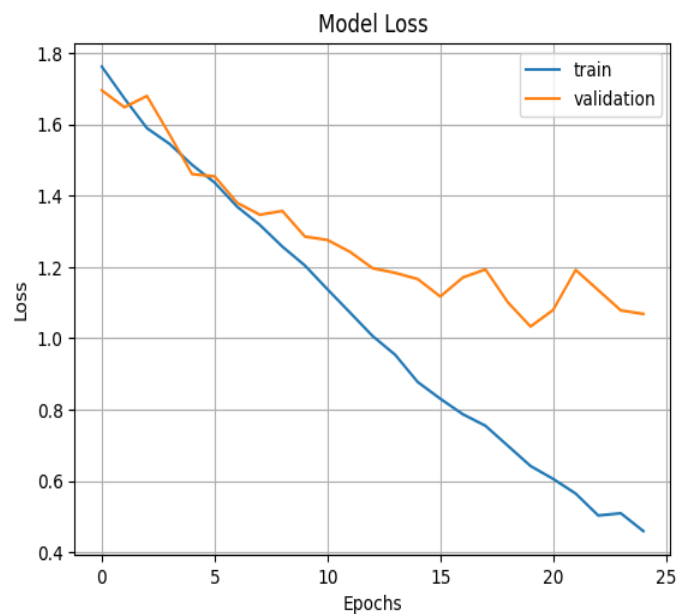
Figure 4.4 shows the accuracy plot where the validation accuracy and train accuracy increases as the number of epochs increases and the loss plot shows that the validation loss and train loss decreases as the number of epochs increases. It also shows the confusion matrix which depicts how many images have been predicted correctly and those numbers are in the diagonal of matrix. Here the predicted label matches the true label. It is also inferred that few number of images of some classes are predicted as different class.

➤ CNN MODEL

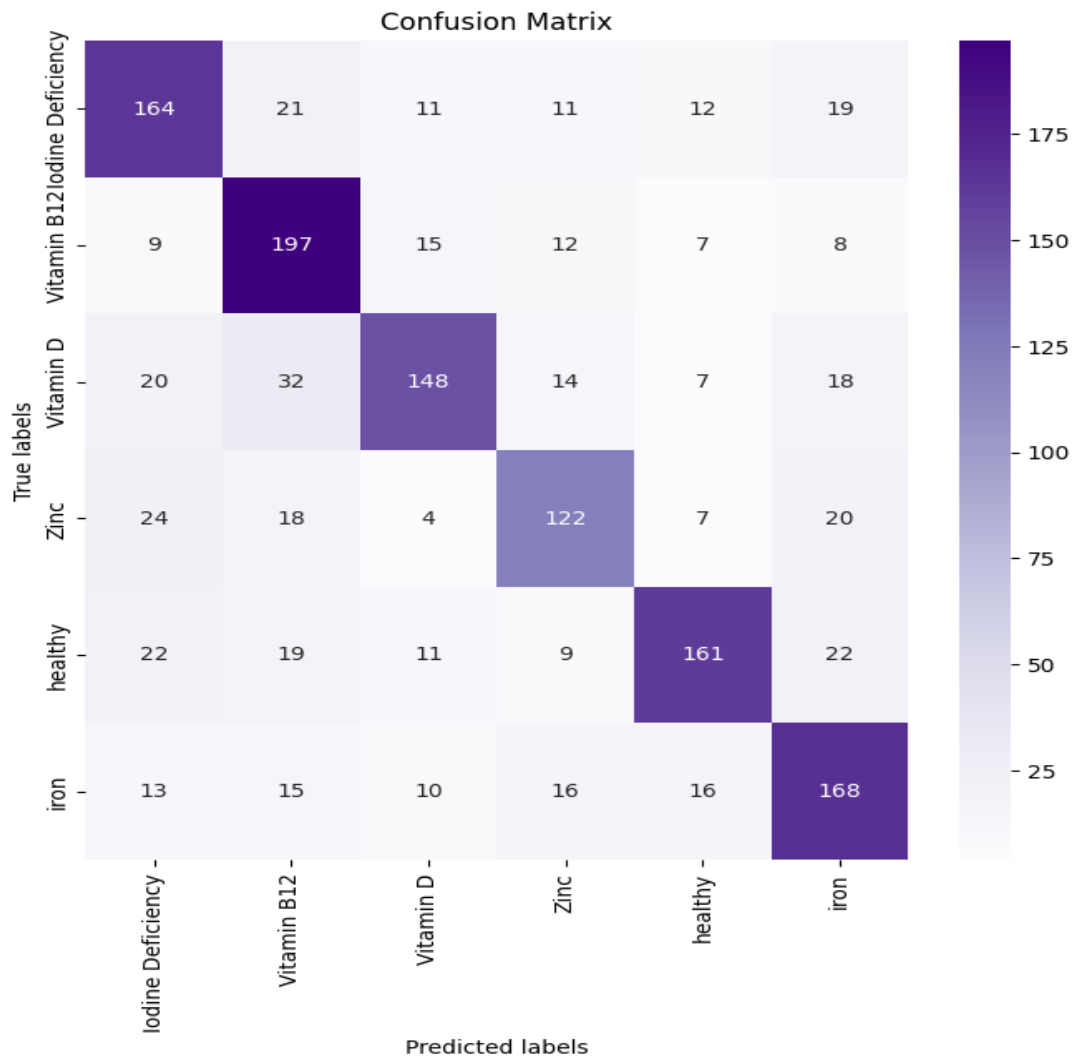
We built a CNN model for feature extraction and classification. This model gives the test accuracy of 68.47% and train accuracy is of 86.10%. The number of epochs used is 20, batch size is of 32, learning rate is 0.001 and Adam optimizer is used here.



(a) Accuracy plot



(b) Loss plot



(c) Confusion matrix

Figure 4.5: Qualitative analysis of CNN model

Figure 4.5 shows the accuracy plot where the validation accuracy and train accuracy increases as the number of epochs increases and the loss plot shows that the validation loss and train loss decreases as the number of epochs increases. It also shows the confusion matrix which depicts how many images have been predicted correctly and those numbers are in the diagonal of matrix. Here the predicted label matches the true label. It is also inferred that few number of images of some classes are predicted as different class.

From all these five models used, ResNet152V2 gave comparatively high accuracy and better performance.

**Table 4.4: Comparative analysis of accuracy
and computational time of the different models**

MODEL	Without augmentation (3310 images)			With augmentation (7000 images)		
	Train Accuracy (%)	Test Accuracy (%)	Time (min)	Train Accuracy (%)	Test Accuracy (%)	Time (min)
RESNET152V2	86.11	74	33.58	96.61	88.94	80.63
INCEPTIONV3	75.59	65.28	62.12	92.46	83.67	49.32
XCEPTION	91.19	70.33	31.01	94.52	83.02	110.4
MOBILENETV2	93.69	69.43	21.13	95.52	83.18	36.42
TRADITIONAL CNN	62.96	43.37	55.12	86.10	68.47	52.13

The above table 4.4 compares the train, test accuracy and the execution time of each model. From this table it is inferred that ResNet152V2 model gave the highest accuracy comparatively. Even though ResNet152v2 and xception has given the highest accuracy, the computational time of both is more when compared to other models. It is also inferred that the accuracy of each model is increased after augmentation. For traditional CNN, we built a model with 5 convolution layers and 5 max pooling layers, that gave comparatively low accuracy in both train and validation.

Table 4.5: Comparative Analysis of Performance metrics

MODEL	DEFICIENCY	PRECISION	RECALL	F1 SCORE	SUPPORT
ResNet152 V2	Iodine	0.92	0.79	0.85	183
	Vitamin B12	0.80	0.80	0.80	168
	Vitamin D	0.80	0.89	0.84	177
	Zinc	0.85	0.68	0.76	141
	Healthy	0.73	0.91	0.81	162
	Iron	0.93	0.90	0.91	171
Inception V3	Iodine	0.80	0.87	0.83	238
	Vitamin B12	0.82	0.80	0.81	248
	Vitamin D	0.86	0.87	0.87	239
	Zinc	0.72	0.85	0.78	195
	Healthy	0.88	0.78	0.83	244
	Iron	0.94	0.85	0.89	238
Xception	Iodine	0.81	0.86	0.84	238
	Vitamin B12	0.86	0.71	0.78	248
	Vitamin D	0.91	0.82	0.87	239
	Zinc	0.79	0.82	0.80	195
	Healthy	0.71	0.91	0.80	244
	Iron	0.94	0.87	0.90	238
Traditional Method	Iodine	0.65	0.69	0.67	238
	Vitamin B12	0.65	0.79	0.72	248
	Vitamin D	0.74	0.62	0.68	239
	Zinc	0.66	0.63	0.64	195
	Healthy	0.77	0.66	0.71	244
	Iron	0.66	0.71	0.68	238

The above table shows that for ResNet 152 V2 model the precision is (93%) high for iron class, recall is (91%) high for healthy class, F1-Score is (91%) high for iron class. For the Inception V3 model the precision is (94%) high for iron class, recall is (87%) high for iodine and vitamin D class, F1-Score is (89%) high for iron class. For the Xception model the precision is (94%) high for iron class, recall is (91%) high for healthy class, F1-Score is (90%) high for iron class. For MobileNet V2 model the precision is (94%) high for iodine class, recall is (94%) high for iron class, F1-Score is (89%) high for healthy class. For Traditional method model the precision is (77%) high for healthy class, recall is (79%) high for Vitamin B12 class, F1-Score is (72%) high for vitamin B12 class and support shows how many number of images are there in validation set for each class. The above comparison is also represented pictorially as below using bar chart representation.

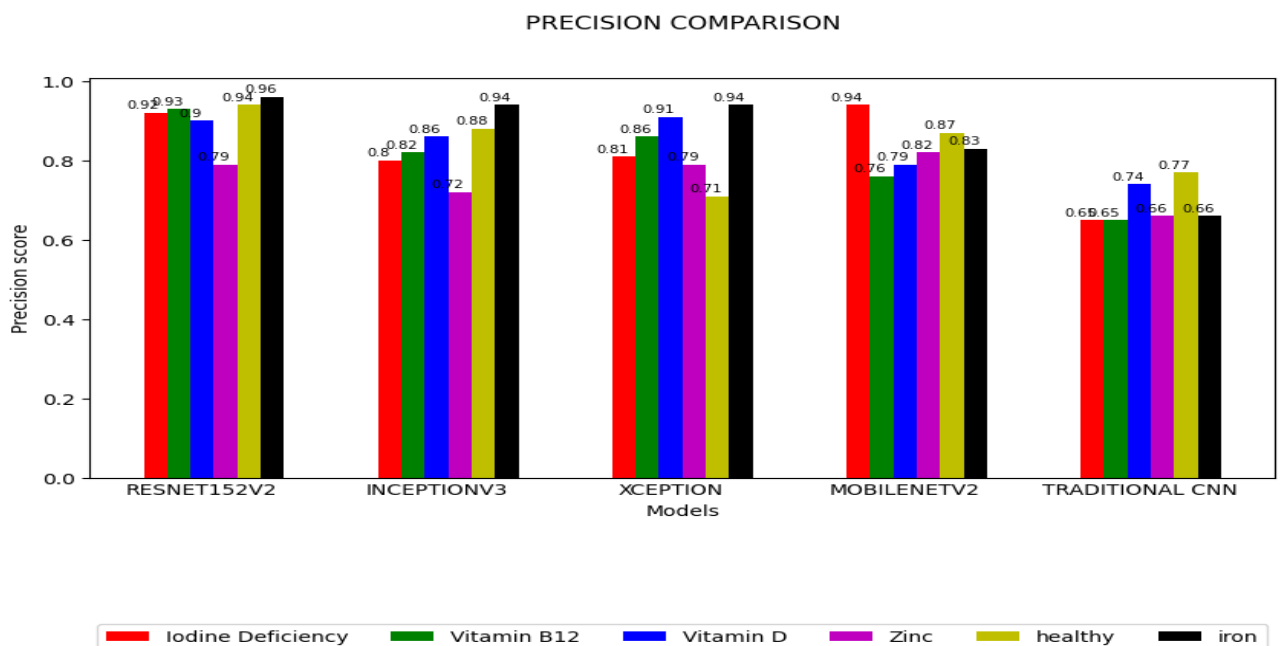


Figure 4.6 Qualitative analysis of Precision of each class

The above figure 4.6 show the precision value of each class with respect to each model. From the bar chart it is inferred that Resnet152v2 model gave overall high

precision value for each class than the other model used. Next to ResNet152V2, MobileNet gave comparatively high precision values.

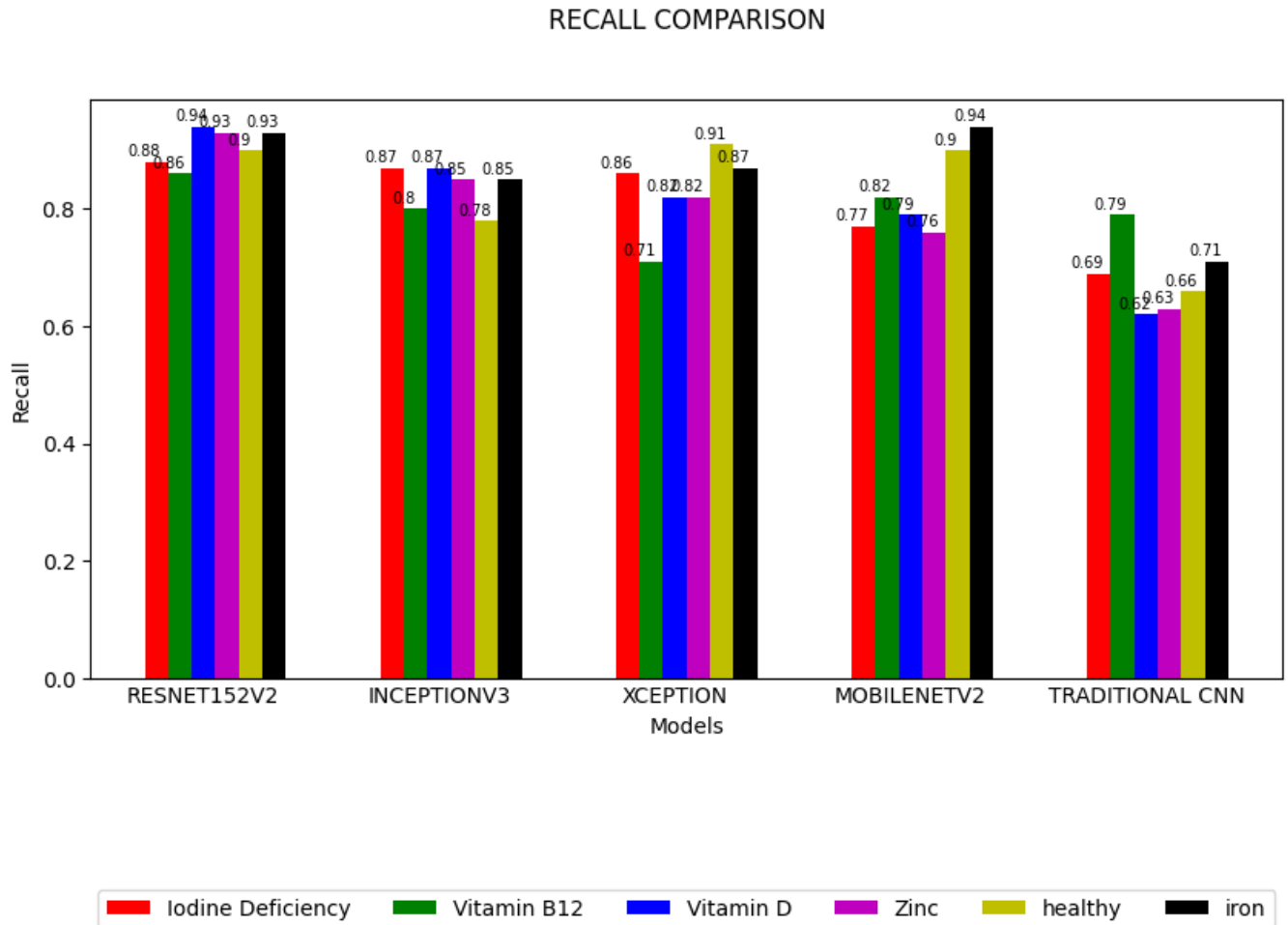


Figure 4.7 Qualitative analysis of Recall of each class

The above figure 4.7 show the Recall value of each class with respect to each model. From the bar chart it is inferred that Resnet152v2 model gave overall high recall value for each class than the other model used. Next to ResNet152V2, MobileNet and Xception models gave comparatively high Recall values.

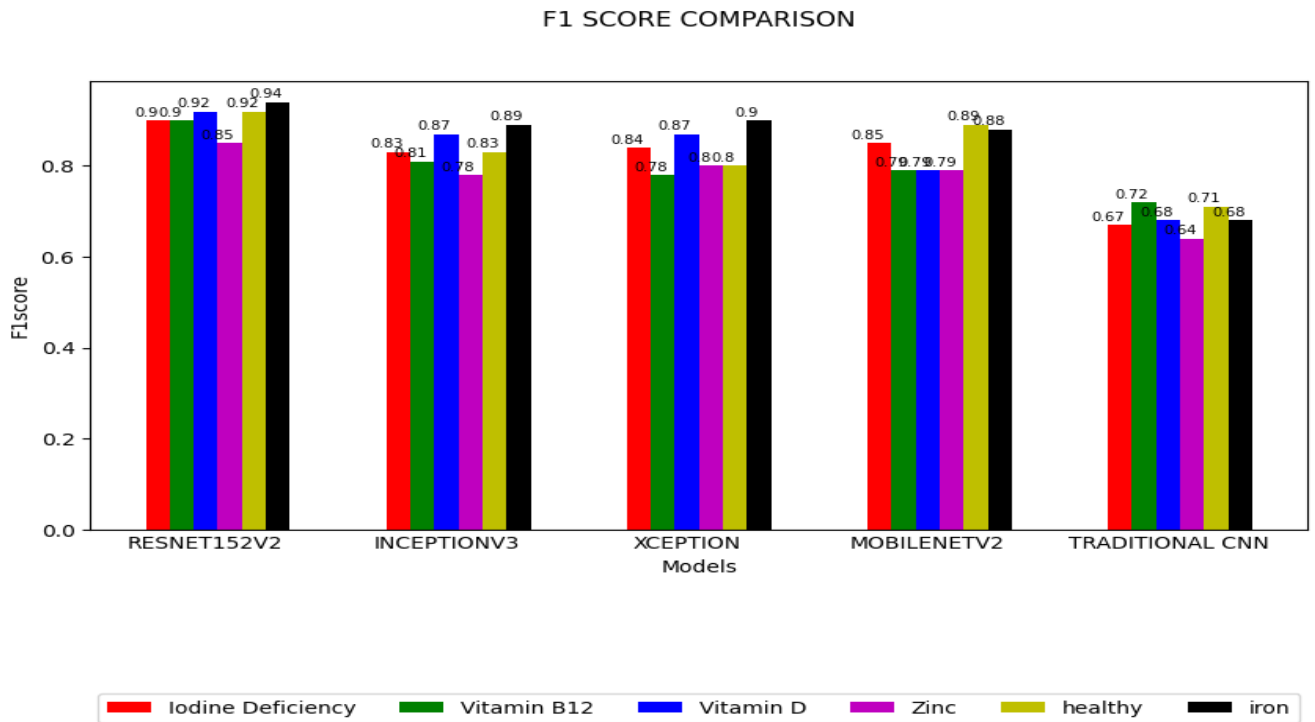


Figure 4.8 Qualitative analysis of F1 score of each class

The above figure 4.8 show the F1 score of each class with respect to each model. From the bar chart it is inferred that Resnet152v2 model gave overall high F1 score for each class than the other model used. Next to ResNet152V2, MobileNet and Xception models gave comparatively high F1 score values.

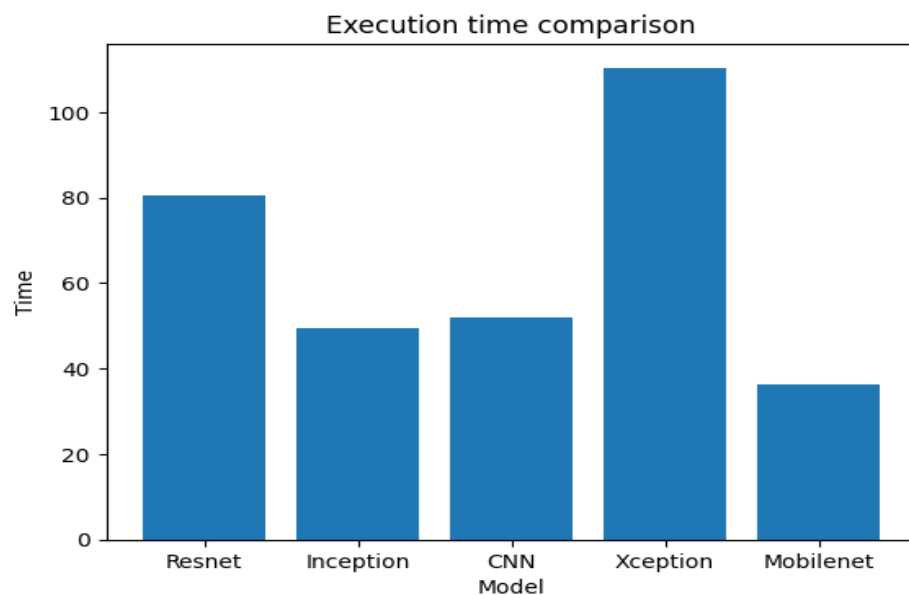
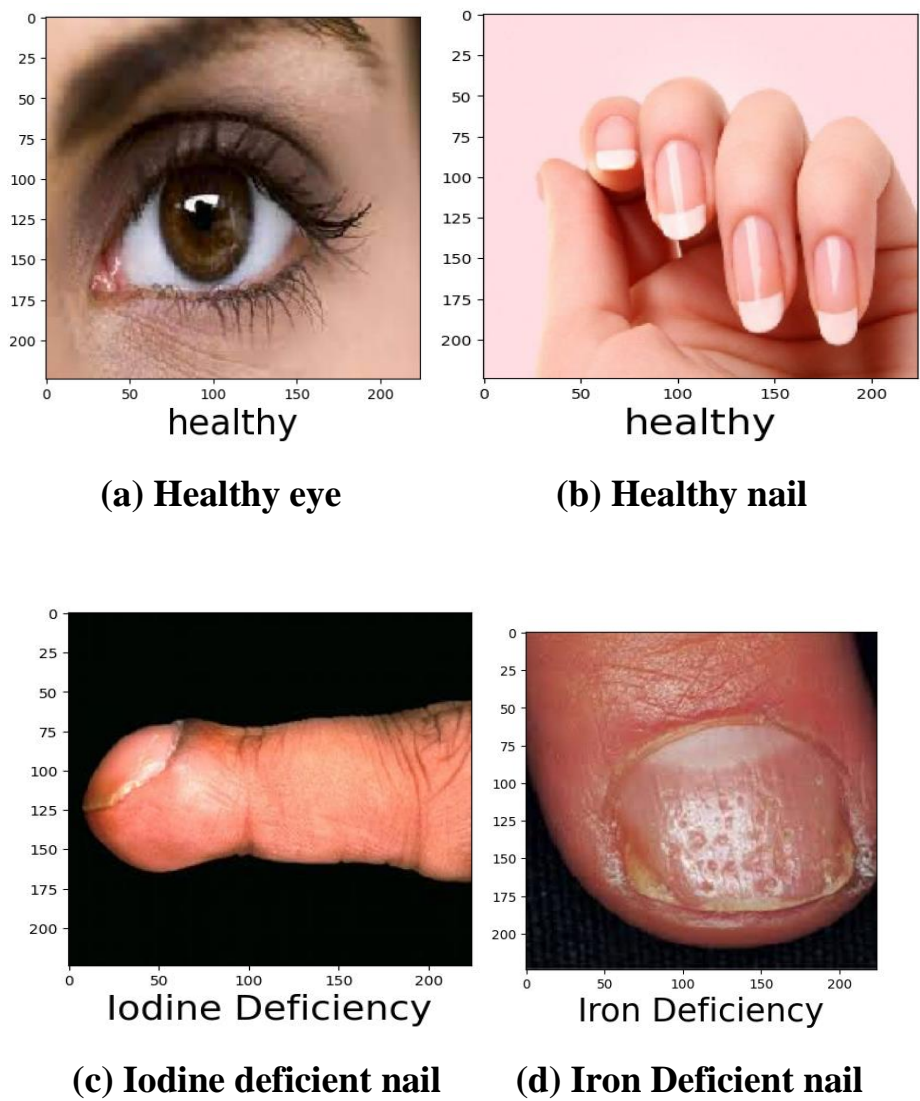


Figure 4.9 Qualitative analysis of Execution time of models.

The above figure 4.9 shows the comparison between execution time and deep learning models. From the bar chart, it is inferred that xception model takes more time for execution. This may be due to the large number of layers in that model. The Xception model took 1 hour 84 minutes to complete the execution. The number of epochs given is 20. From overall execution time of that model, the first epoch of the 20 epochs took more time to complete the execution.

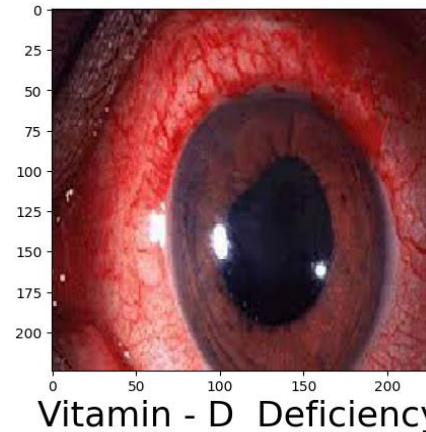
4.2.4 Image Prediction using Deep learning models for nail dataset

Image Prediction is nothing but it is the output of an algorithm after it has been trained on a historical dataset and applied to new data when forecasting the likelihood of a particular outcome.

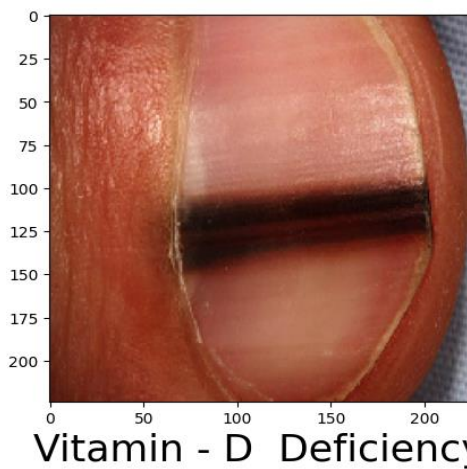




(e) Vitamin B12 deficient nail



(f) Vitamin-D deficient eye



(g) Vitamin-D deficient nail



(h) Zinc deficient nail

Figure 4.10: Image prediction using human datasets

The above figure 4.10 shows the outcomes that are predicted as Healthy eye, Healthy nail, Iodine, Iron, Vitamin B12, Vitamin-D deficiency in eye and nail, Zinc deficiency using Resnet 152 V2 model in human dataset. Similarly the deficiencies are predicted using other models such as Inception V3, Xception, MobileNet V2, CNN method.



4.3 HIDDEN HUNGER DETECTION USING PLANT IMAGES




Micro nutrient deficiency can be identified from different visible changes in leaves of the plants. Totally 1560 images were used for deficiency detection of which after augmentation 70% is used as training data, 20% is used as validation data, 10% is used as test data. Dataset has 5 classes of micronutrient deficiency. Each class has

images of specific micronutrient deficiency. The five classes are Boron, Iron, Manganese, Zinc and Healthy. The dataset consists of banana leaf images of various categories like Musa acuminata (Dwarf Cavendish), Robusta, Rasthali, Poovali, Poovan, Monthan, Elakkibale. The dataset is taken from Mendeley. The link for the dataset is given below.

Mendeley-<https://data.mendeley.com/datasets/7vpdrbdkd4>

Table 4.6: Different classes in leaf dataset and their sample images.

SAMPLE IMAGE	CLASS NAME	VISIBLE SYMPTOM OF DEFICIENCY	NO. OF IMAGES
	Boron Deficiency	It affects the youngest leaves. Leaves have become chlorotic from their margin and curled inwards also lead to necrosis on upper leaves. The necrosis become lighter brown	100
	Iron Deficiency	Leaves become yellowish green and later become totally yellow. Midrib and leaf vein remain green also fade out with severe deficiency	86

	Manganese Deficiency	It affect the younger leaves. The leaves become lighter green with slight deficiency. For severe case chlorosis appear in intercoastal.	24
	Zinc Deficiency	Zones at leaf margin chlorotic zones in the intercoastal areas which are spread over the whole leaf and leaf give mottled appearance. Midrib does not remain green.	400
	Healthy	—	950

The above table shows the different classes of leaf images and its corresponding micro nutrient deficiency. This table also shows how many number of images are there in each class and also describes the visible deficiency symptoms in leaf. The images used for training are of dimension $255 \times 255 \times 3$.

4.3.1. Data augmentation





As similar to what has been done with nail dataset, the first step is to augment the dataset. The following techniques are used for augmentation in the specified proportion, Zoom-20%, Flip-20%, Random distortion-30%, Random Brightness-10%. Initially, there are 1560 images and. It is augmented to 3400 images and the performance of the model is evaluated. Again, in order to improve the performance of model, the images are augmented to 5000 and again performance of model is evaluated. It is inferred here that the model performed better with 5000 images than with 3400 images.

Table 4.7: Data Augmentation result

Classes	No of images before augmentation (1560)	No of images after Augmentation (3700)	No of images after Augmentation (5000)
Boron	100	800	1200
Iron	86	750	1000
Manganese	24	150	700
Zinc	400	800	900
Healthy	950	1200	1200

The above table shows that number of images before augmentation is 1560 and after augmentation it gets increased to 3400 images. When it is augmented for the second time it results with 5000 images. The table also shows number of images in each class before and after augmentation.

Table 4.8: Augmentation techniques

Augmentation technique	Sample image	Probability of augmentation
Zoom		30%
Flip top-bottom		30%
Random brightness		20%
Random distortion		10%

The above table shows the different techniques used for augmentation. The sample images for each technique is also given in the table. This table also gives the probability of augmentation which means from the set of images 30% of them will be zoomed and 30% will be flipped and 20% will be subjected to brightness change

and 10% will be randomly distorted.

4.3.2. Splitting the dataset

After augmentation, data is split into train and test in the ratio of 7:2:1, where 70% is training data and 20% is validation data and 10% is the test data. After augmenting dataset to 3400 images, there are 2380 images in train dataset, 680 images in validation dataset and 340 images in test data. When augmenting the same dataset to 5000 images, there are 3500 images in train dataset and 1000 images in validation dataset and 500 images in test dataset.

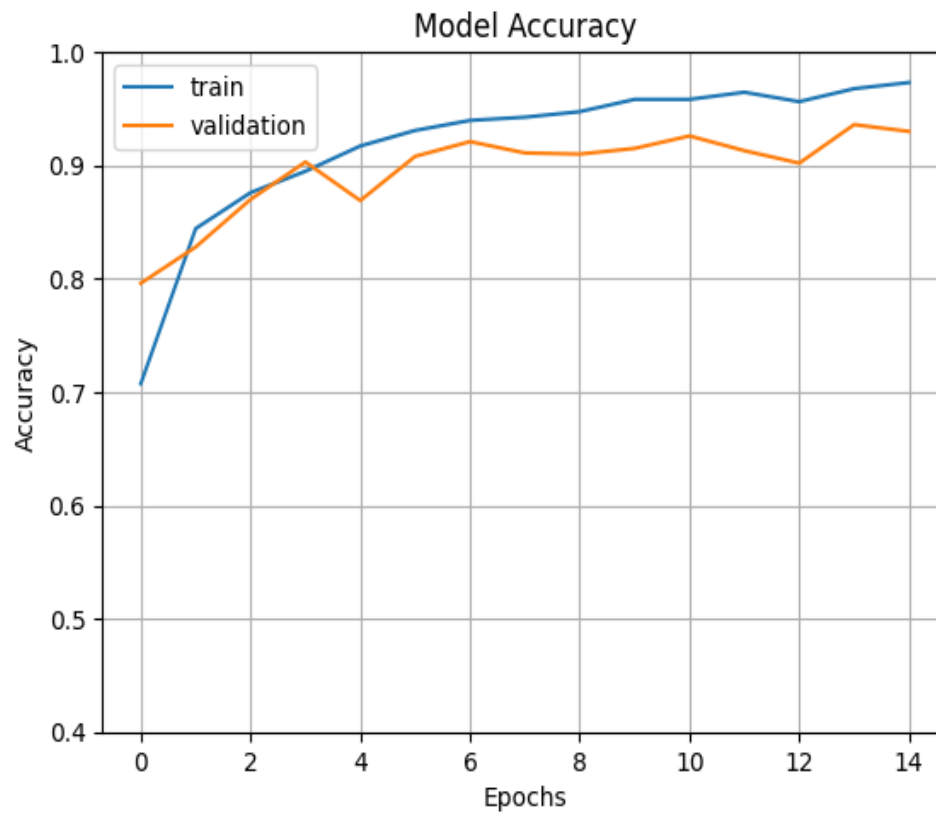
4.3.3. Deep learning models

We used four different models for feature extraction and classification.

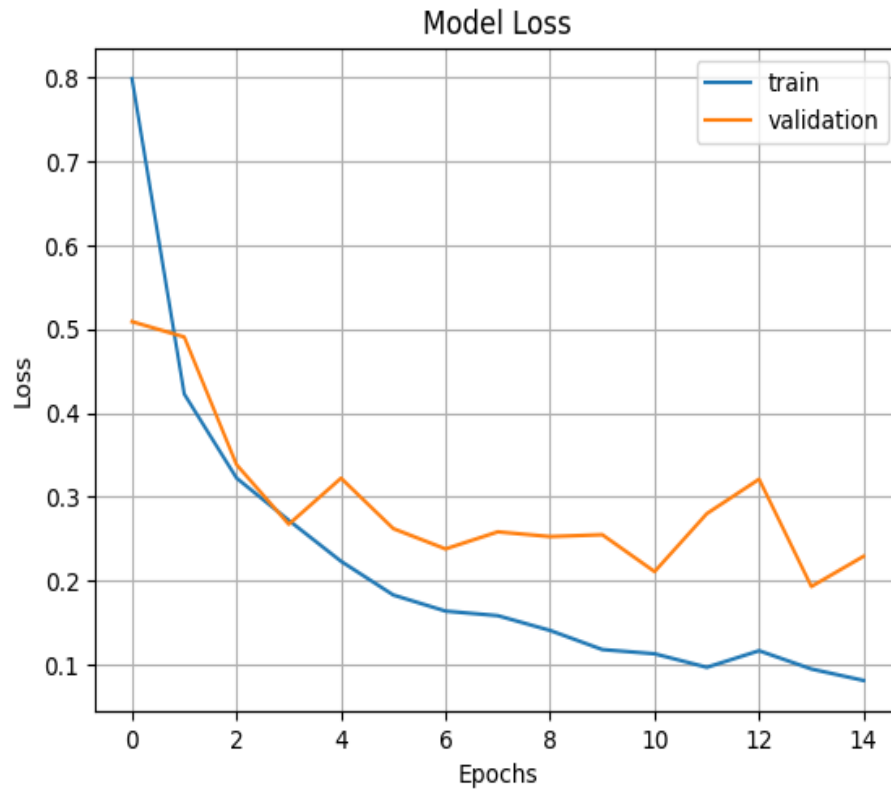
- RESNET 152 V2
- Inception V3
- Xception
- MobileNet V2
- Traditional Method

➤ RESNET 152 V2

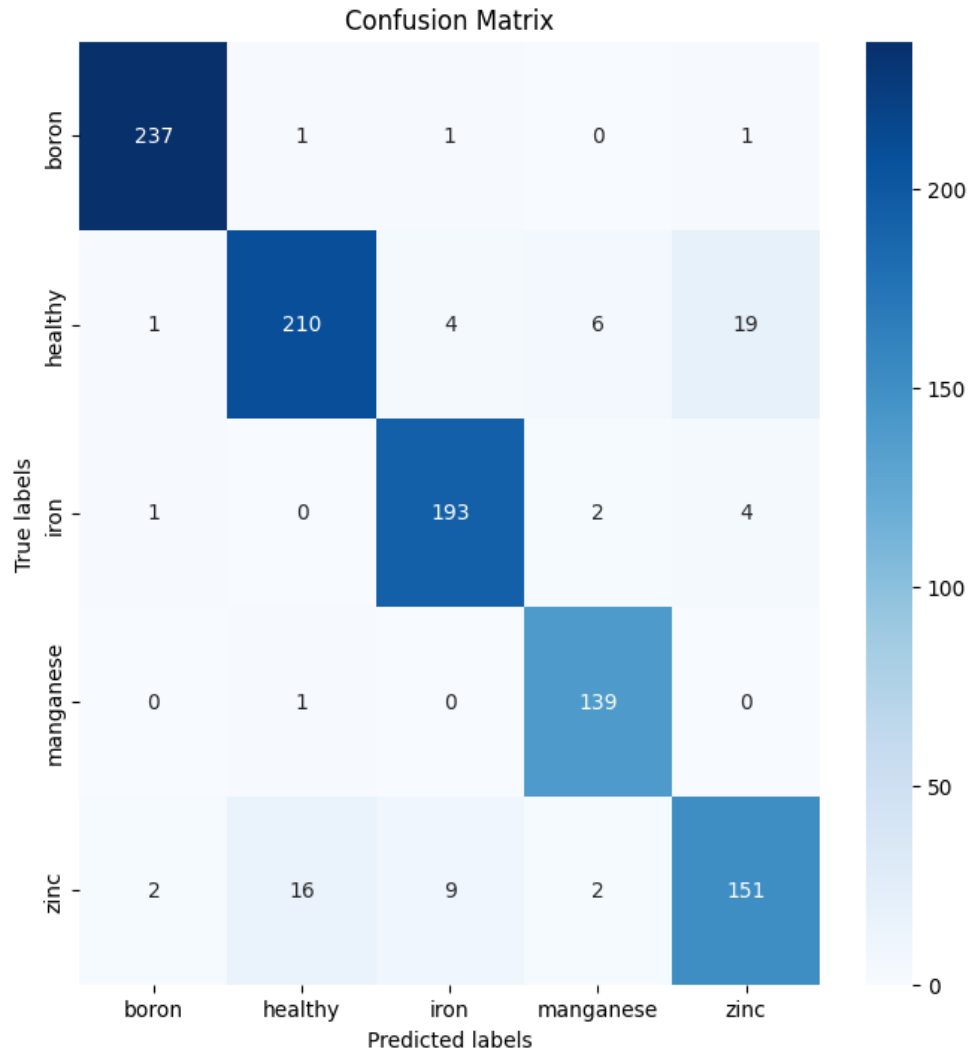
We use ResNet 152 V3 model for feature extraction and classification. This model gives the test accuracy of 83.33% and train accuracy is of 92.29%. The number of epochs used is 20, batch size is of 32, learning rate is 0.001 and adam optimizer is used here. This model gave the highest accuracy compared to other models. The reason may be that this architecture has 152 number of convolution layers so the learn even complex data more accurately.



(a) Accuracy plot



(b) Loss plot



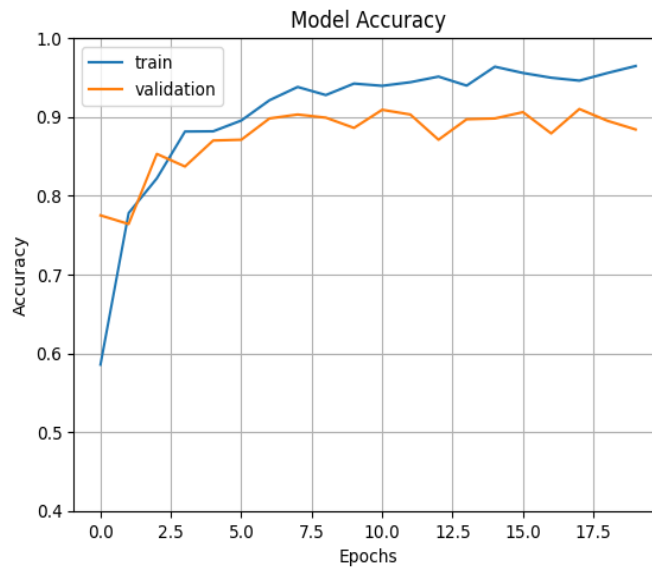
(c) Confusion Matrix

Figure 4.11: Qualitative analysis of ResNet 152 V2

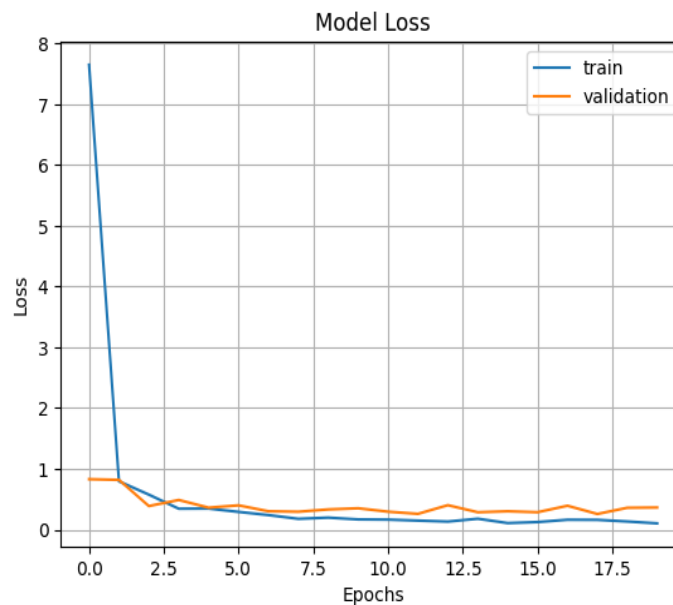
Figure 4.11 shows the accuracy plot where the validation accuracy and train accuracy increases as the number of epochs increases and the loss plot shows that the validation loss and train loss decreases as the number of epochs increases. It also shows the confusion matrix which depicts how many images have been predicted correctly and those numbers are in the diagonal of matrix. Here the predicted label matches the true label. It is also inferred that few number of images of some classes are predicted as different class.

➤ INCEPTION V3

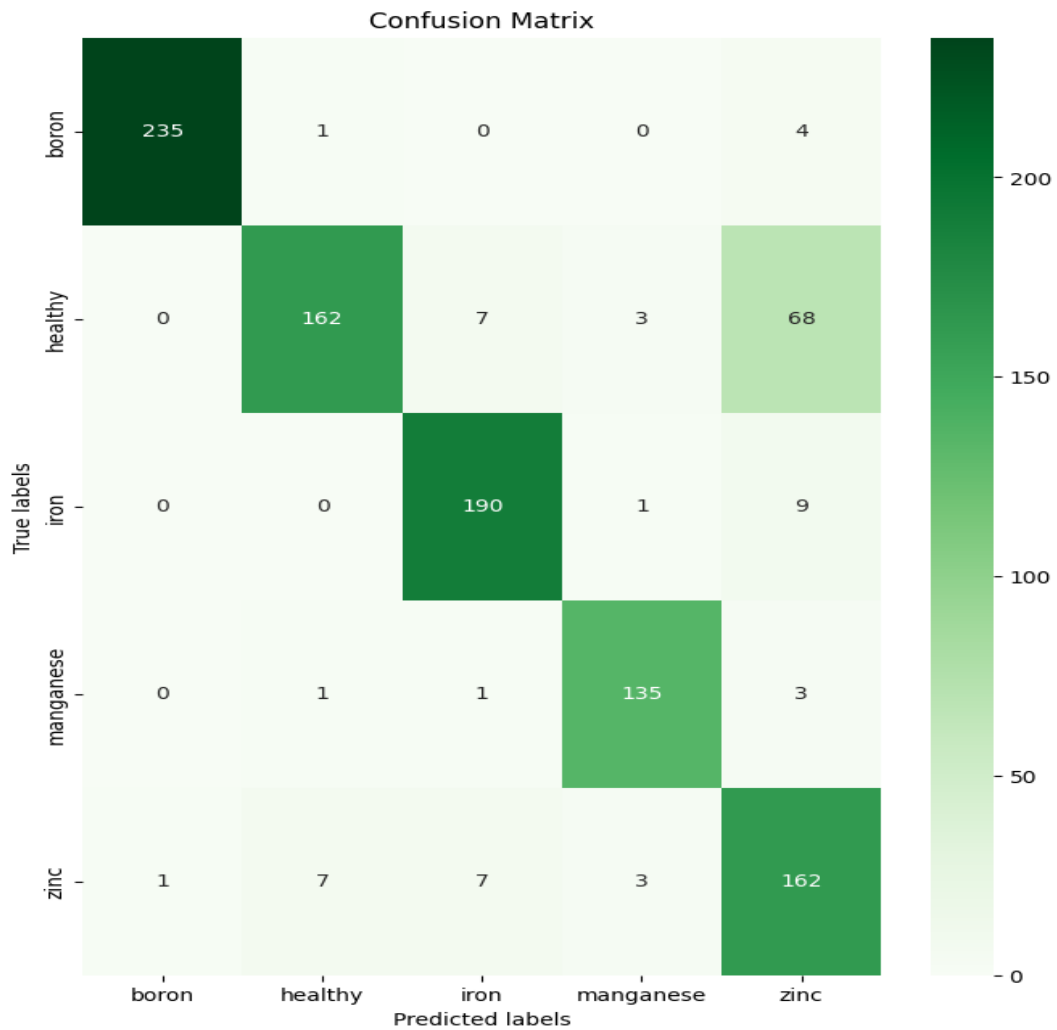
We use Inception V3 model for feature extraction and classification. This model gives the test accuracy of 83.67% and train accuracy is of 92.46%. The number of epochs used is 20, batch size is of 32, learning rate is 0.001 and adam optimizer is used here.



(a) Accuracy plot



(b) Loss plot



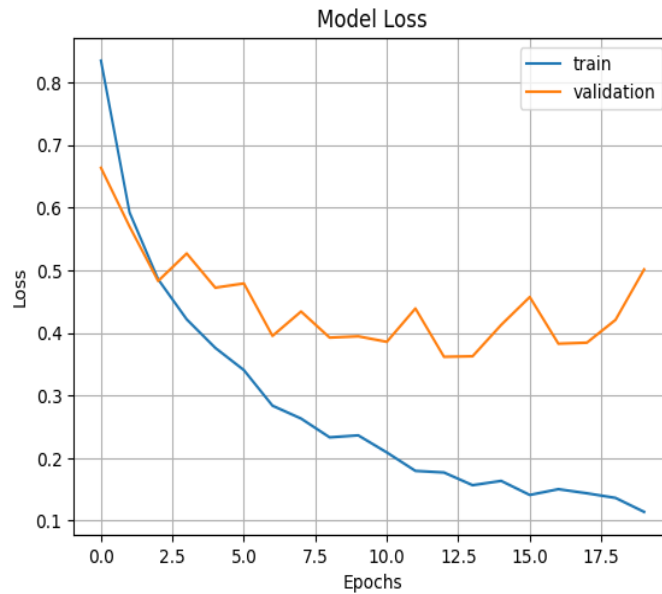
(c) Confusion Matrix

Figure 4.12: Qualitative analysis of Inception V3

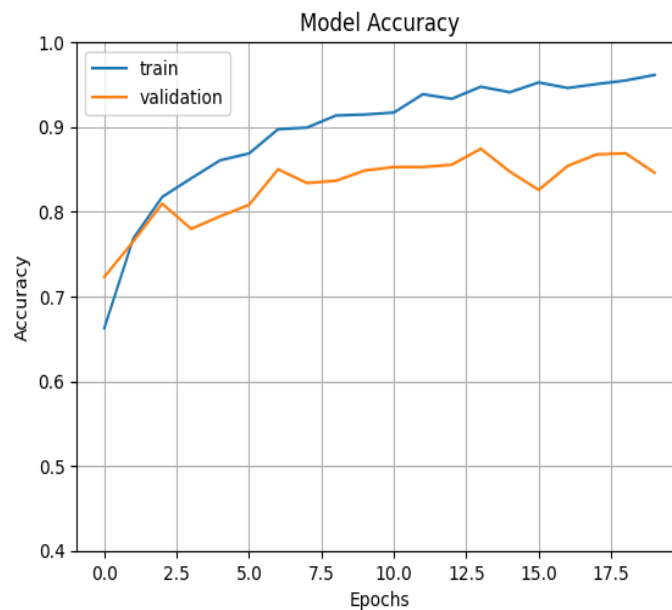
Figure 4.12 shows the accuracy plot where the validation accuracy and train accuracy increases as the number of epochs increases and the loss plot shows that the validation loss and train loss decreases as the number of epochs increases. It also shows the confusion matrix which depicts how many images have been predicted correctly and those numbers are in the diagonal of matrix. Here the predicted label matches the true label. It is also inferred that few number of images of some classes are predicted as different class.

➤ XCEPTION

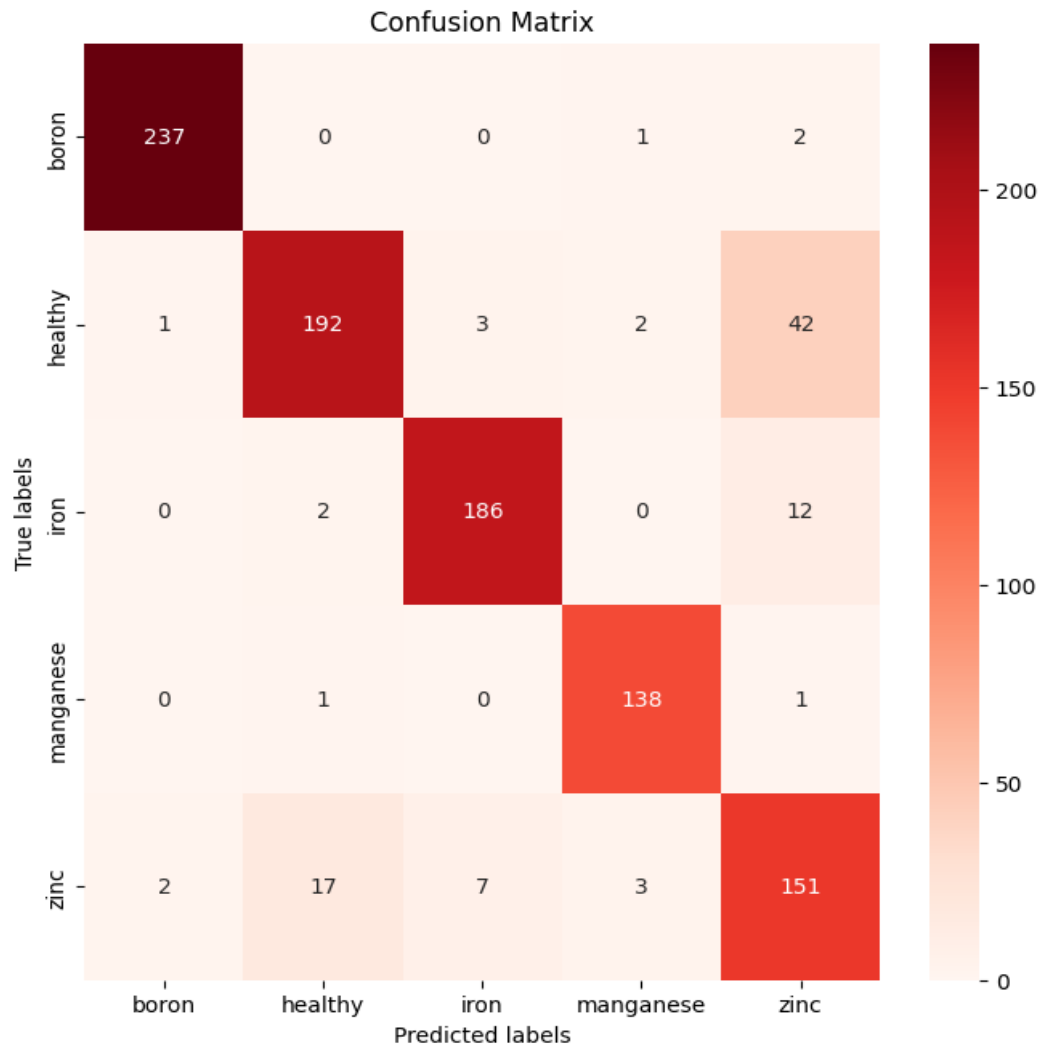
We use Xception model for feature extraction and classification. This model gives the test accuracy of 83.67% and train accuracy is of 92.46%. The number of epochs used is 20, batch size is of 32, learning rate is 0.001 and adam optimizer is used here.



(a) Accuracy plot



(b) Loss plot



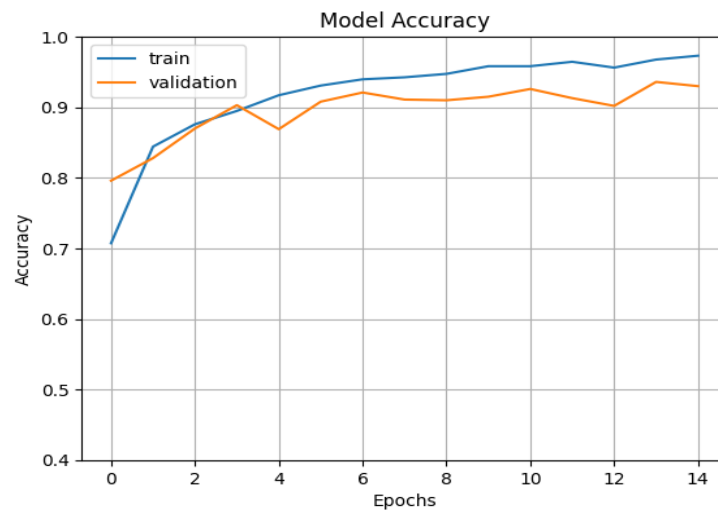
(c) Confusion Matrix

Figure 4.13: Qualitative analysis of Xception

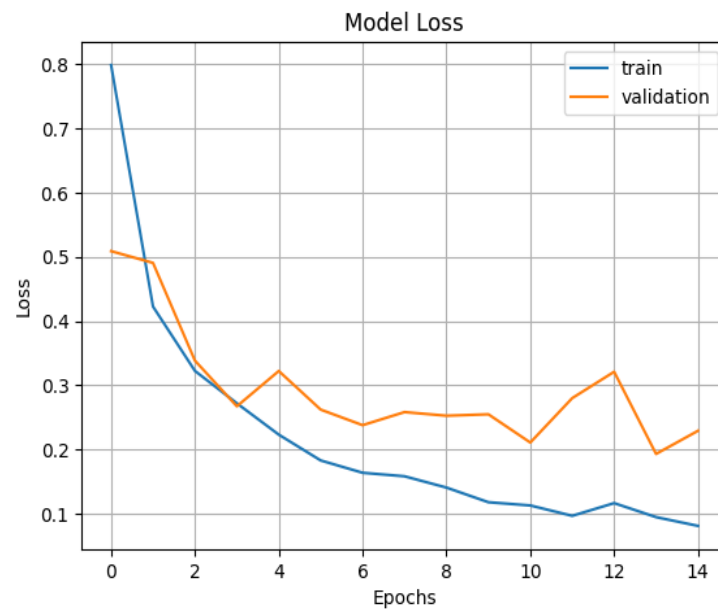
Figure 4.13 shows the accuracy plot where the validation accuracy and train accuracy increases as the number of epochs increases and the loss plot shows that the validation loss and train loss decreases as the number of epochs increases. It also shows the confusion matrix which depicts how many images have been predicted correctly and those numbers are in the diagonal of matrix. Here the predicted label matches the true label. It is also inferred that few number of images of some classes are predicted as different class.

➤ MOBILENET V2

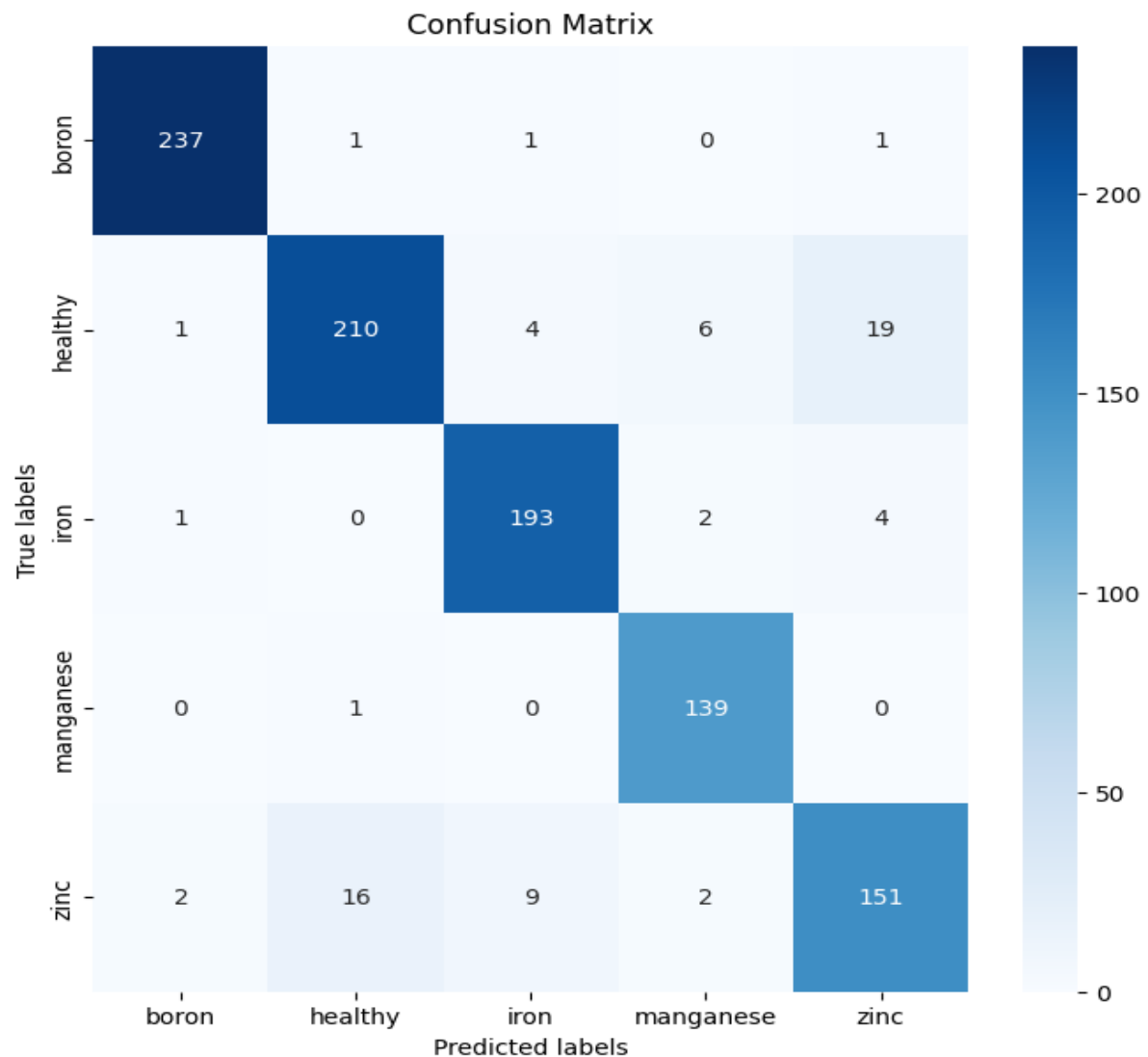
We use MobileNet V2 model for feature extraction and classification. This model gives the test accuracy of 83.67% and train accuracy is of 92.46%. The number of epochs used is 20, batch size is of 32, learning rate is 0.001 and adam optimizer is used here.



(a) Accuracy Plot



(b) Loss plot



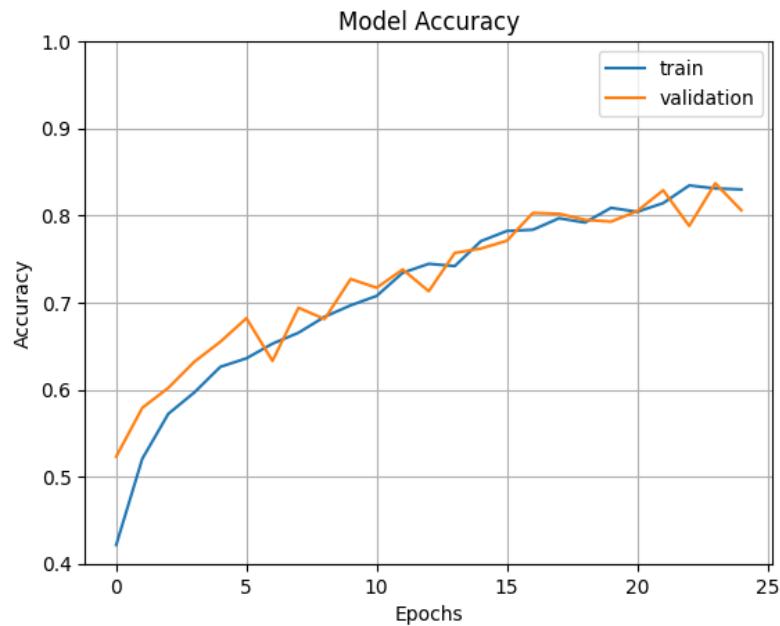
(c) Confusion Matrix

Figure 4.14: Qualitative analysis of MobileNet

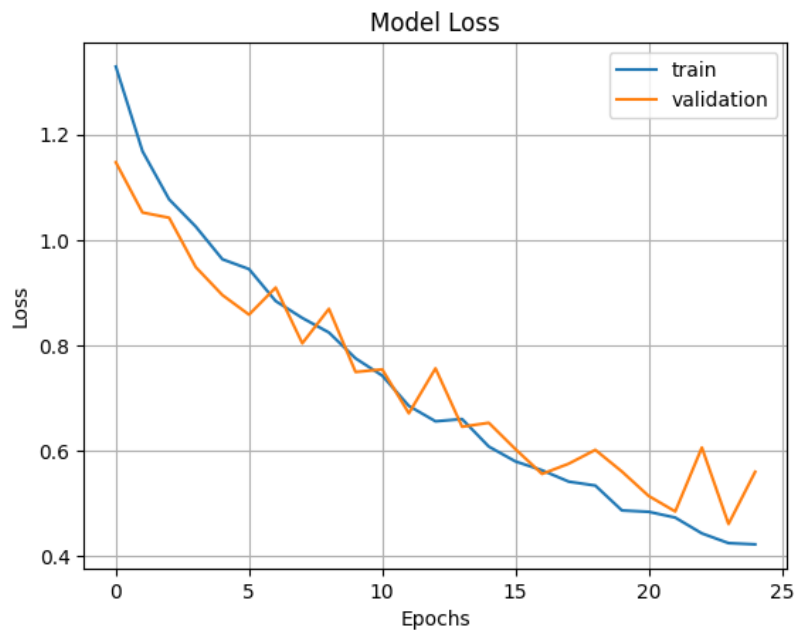
Figure 4.14 shows the accuracy plot where the validation accuracy and train accuracy increases as the number of epochs increases and the loss plot shows that the validation loss and train loss decreases as the number of epochs increases. It also shows the confusion matrix which depicts how many images have been predicted correctly and those numbers are in the diagonal of matrix. Here the predicted label matches the true label. It is also inferred that few number of images of some classes are predicted as different class.

➤ CNN MODEL

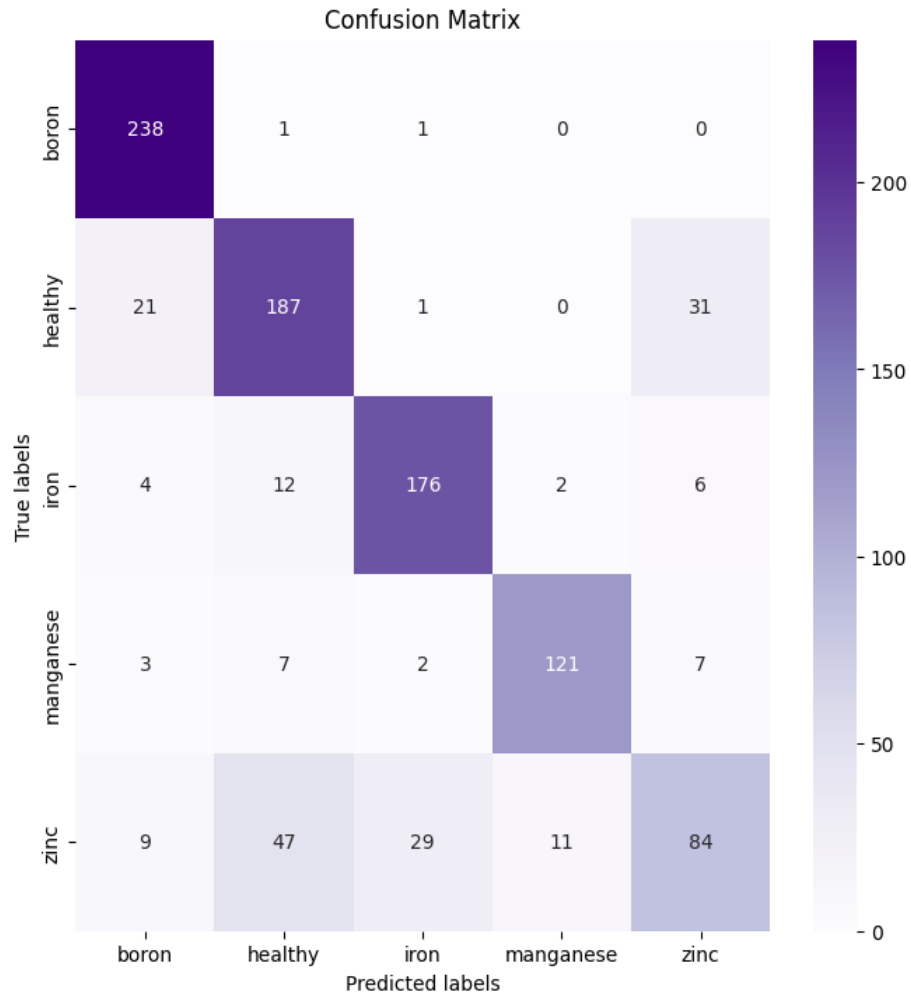
We built a CNN model for feature extraction and classification. This model gives the test accuracy of 83.67% and train accuracy is of 92.46%. The number of epochs used is 20, batch size is of 32, learning rate is 0.001 and adam optimizer is used here.



(a) Accuracy plot



(b) Loss plot



(c) Confusion Matrix

Figure 4.15: Qualitative analysis of Traditional method

Figure 4.15 shows the accuracy plot where the validation accuracy and train accuracy increases as the number of epochs increases and the loss plot where the validation loss and train accuracy decreases as the number of epochs increases. It also shows the confusion matrix which depicts the numbers in the diagonal of matrix and shows how many are predicted correctly so that the predicted label matches the true label and also shows that few number of images of some classes that are predicted as different class.

From all these five models used, ResNet152V2 gave comparatively high accuracy and better performance.

Table 4.9: Comparative analysis of accuracy and computational time of the models

MODEL	Without augmentation (1560 images)			With augmentation (5000 images)		
	Train Accuracy (%)	Test Accuracy (%)	Time (min)	Train Accuracy (%)	Test Accuracy (%)	Time (min)
RESNET152V2	97.03	88.38	40.34	97.31	93.00	17.4
INCEPTIONV3	94.83	84.19	33.7	96.80	90.40	60.38
XCEPTION	96.14	84.59	37.26	96.57	91.40	49.19
MOBILENETV2	93.69	69.4	21.13	97.20	92.30	60.00
TRADITIONAL CNN	81.89	80.27	27.13	83.00	80.60	48.39

The above table compares the train, test accuracy and the execution time of each model. From this table it is inferred that ResNet152V2 and MobileNetV2 gave the highest accuracy comparatively. Even though MobileNetV2 and InceptionV3 has given high accuracy, the computational time of both is more when compared to other models. It is also inferred that the accuracy of each model is increased after augmentation. For traditional CNN, we built a model with 5 convolution layers and 5 max pooling layers, that gave comparatively low accuracy in both train and validation.

Table 4.10: Comparative Analysis of Performance metrics

MODEL	DEFICIENCY	PRECISION	RECALL	F1 SCORE	SUPPORT
ResNet 152 V2	Boron	0.98	0.99	0.99	240
	Manganese	0.93	0.99	0.96	140
	Zinc	0.86	0.84	0.85	180
	Healthy	0.92	0.88	0.90	240
	Iron	0.93	0.96	0.95	200
Inception V3	Boron	1.00	0.98	0.99	240
	Manganese	0.95	0.96	0.96	140
	Zinc	0.66	0.90	0.76	180
	Healthy	0.95	0.68	0.79	240
	Iron	0.93	0.95	0.94	200
MobileNet V2	Boron	0.98	0.99	0.99	240
	Manganese	0.93	0.99	0.96	140
	Zinc	0.86	0.84	0.85	180
	Healthy	0.92	0.88	0.90	240
	Iron	0.93	0.96	0.95	200
Xception	Boron	0.99	0.99	0.99	240
	Manganese	0.96	0.99	0.97	140
	Zinc	0.73	0.84	0.78	180
	Healthy	0.91	0.80	0.85	240
	Iron	0.95	0.93	0.94	200
Traditional Method	Boron	0.87	0.99	0.92	240
	Manganese	0.90	0.86	0.88	140
	Zinc	0.66	0.47	0.55	120
	Healthy	0.74	0.78	0.76	240
	Iron	0.84	0.88	0.86	200

The above table shows that for ResNet 152 V2 model the precision is (98%) more for boron class, recall is (99%) high for boron and manganese class, F1-Score is (99%) high for boron class. For the Inception V3 model the precision is (100%) high for boron class, recall is (98%) high for boron class, F1-Score is (99%) high for boron class. For the MobileNet V2 model the precision is (98%) high for boron class, recall is (99%) high for boron and manganese class, F1-Score is (99%) high for boron class. For the Xception model the precision is (99%) high for boron class, recall is (99%) high for boron and manganese class, F1-Score is (99%) high for boron class. For the traditional method the precision is (90%) high for manganese class, recall is (99%) high for boron class, F1-Score is (92%) high for boron class and support shows how many number of images are there in validation set for each class. The above comparison is also represented pictorially as below using bar chart representation.

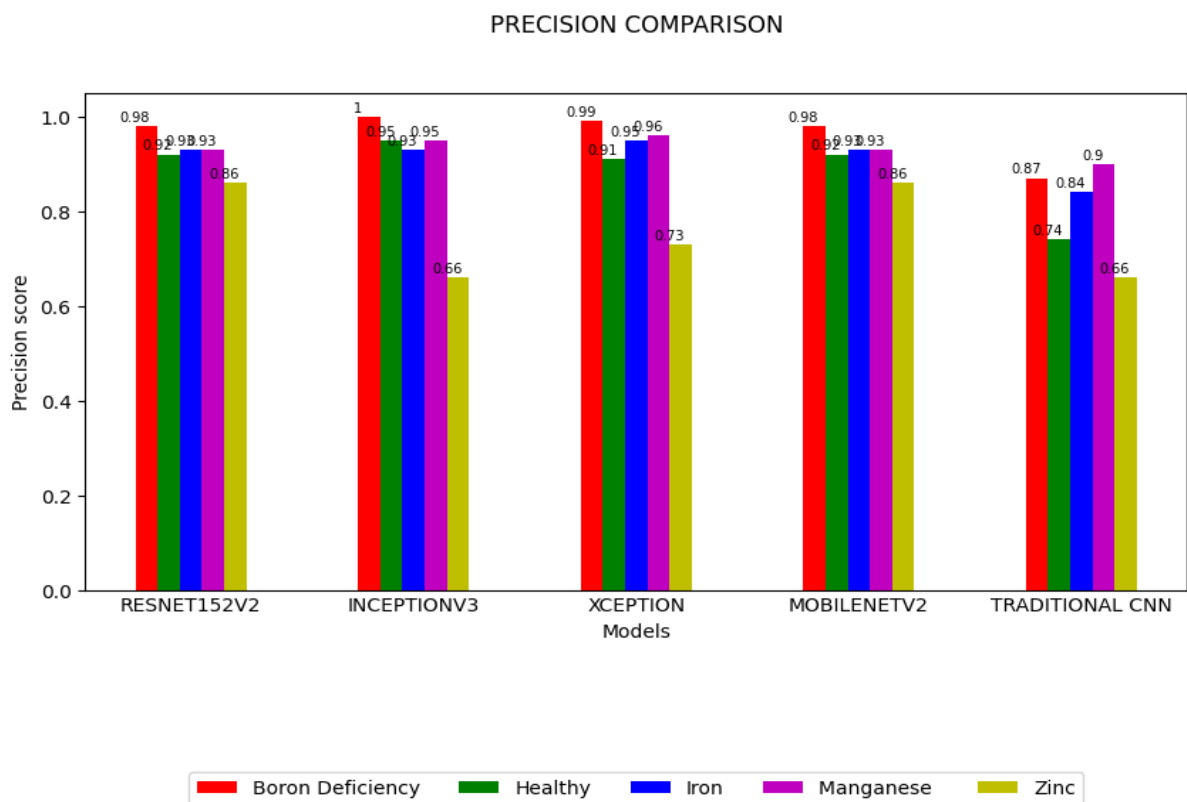


Figure 4.16: Comparative Analysis of Precision of each class

The above figure 4.16 shows the precision value of each class with respect to each model. From the bar chart it is inferred that Resnet152v2 model gave overall high precision value for each class than the other model used. The precision value of Boron class is seemed to be high for all the model used.

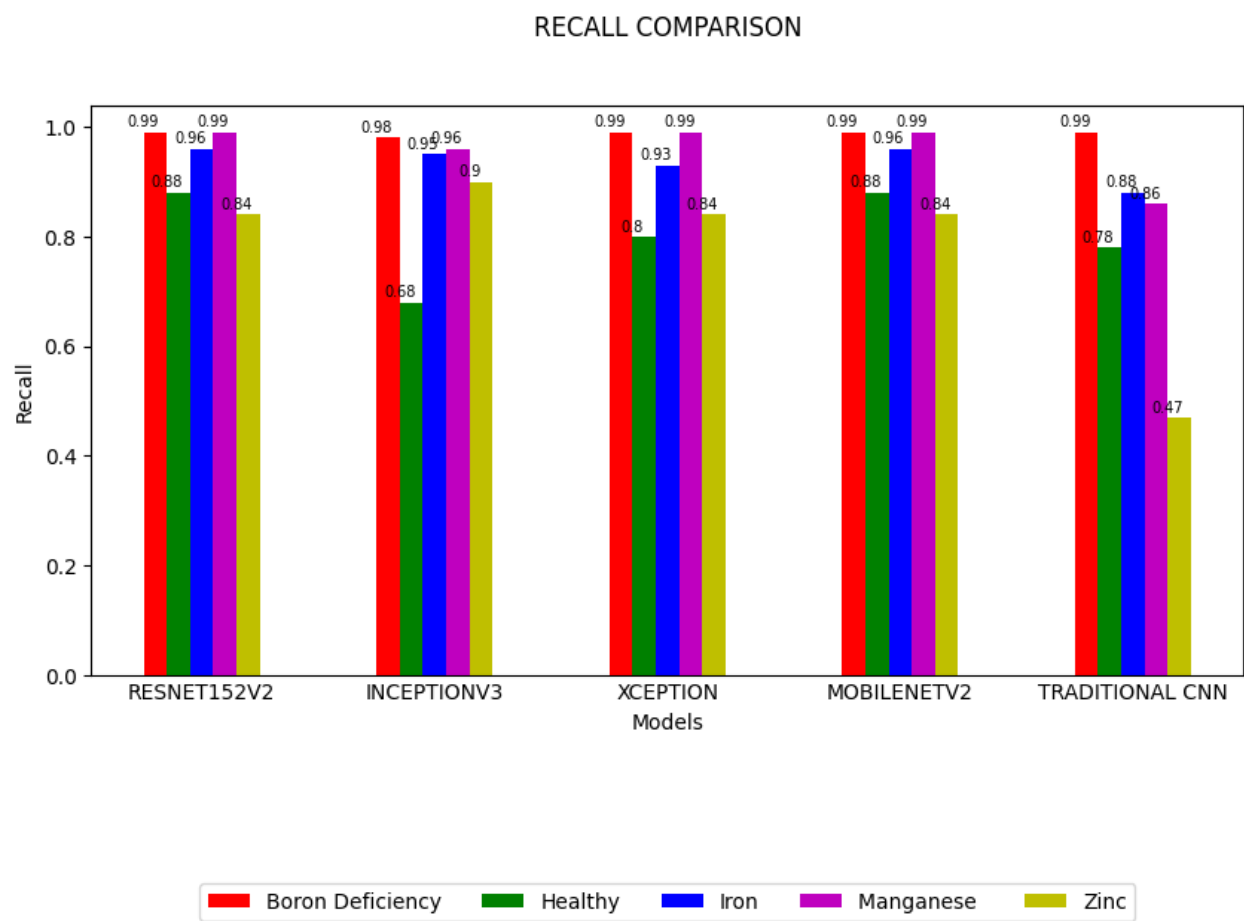


Figure 4.17: Comparative Analysis of Recall values of each class

The above figure 4.17 shows the recall value of each class with respect to each model. From the bar chart it is inferred that Resnet152v2 model gave overall high recall value for each class than the other model used. The recall value of Boron class is seemed to be high for all the model used.

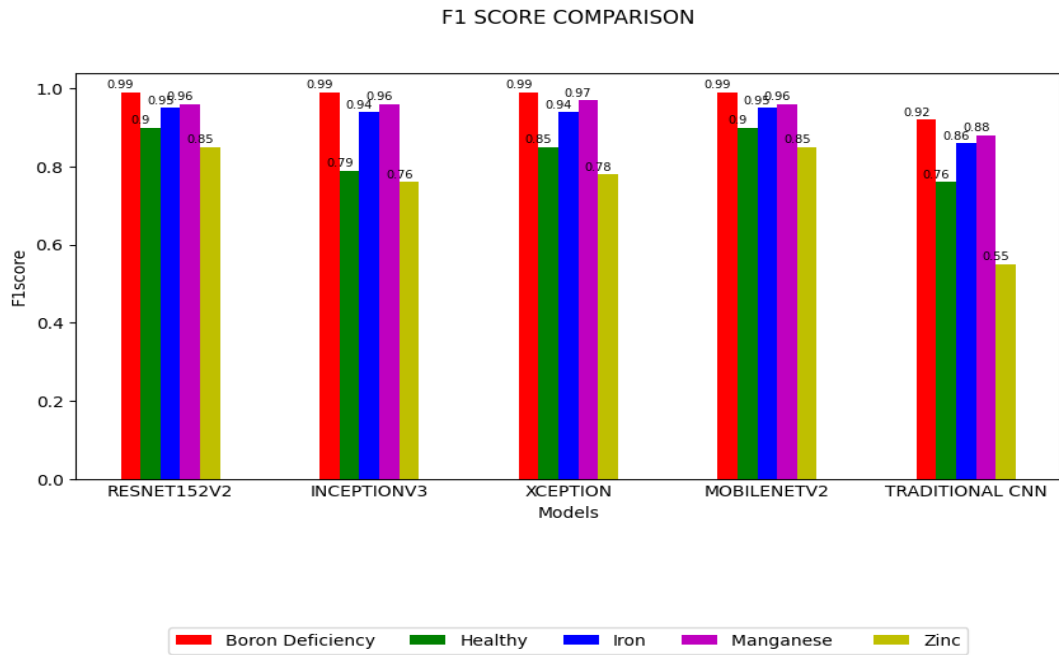


Figure 4.18: Comparative Analysis of F1 score of each class

The above figure 4.18 shows the F1 score of each class with respect to each model. From the bar chart it is inferred that Resnet152v2 model gave overall high F1 score for each class than the other model used. The F1 score of Boron class is seemed to be high for all the model used.

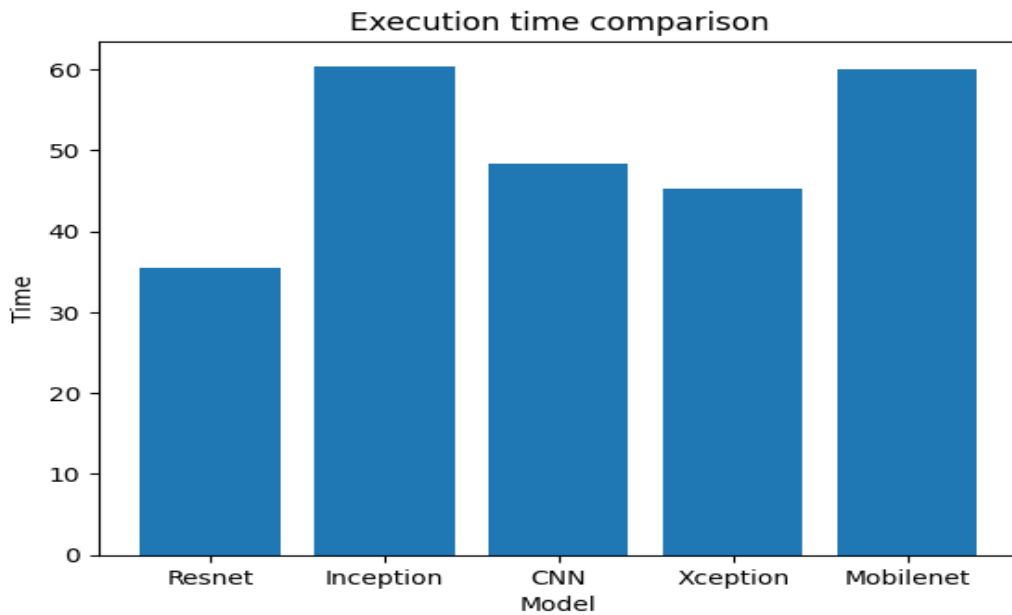
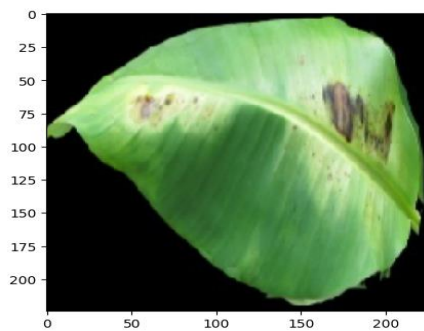


Figure 4.19: Comparative Analysis of Execution time of models.

The above figure 4.19 shows the comparison between execution time and deep learning models. From the bar chart, it is inferred that InceptionV3 and MobileNet model takes more time for execution. This may be due to the large number of layers in that model. The Inception model took 1 hour 38 seconds and MobileNet took 1 hour to complete the execution. The number of epochs given is 20.

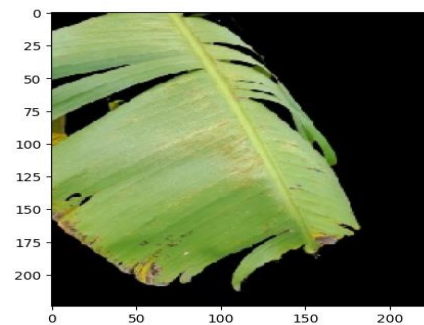
4.3.4 Image Prediction using Deep learning models for leaf dataset

Image Prediction is nothing but it is the output of an algorithm after it has been trained on a historical dataset and applied to new data when forecasting the likelihood of a particular outcome.



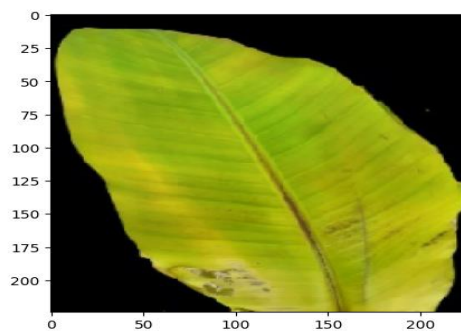
Manganese

(a) Manganese deficiency



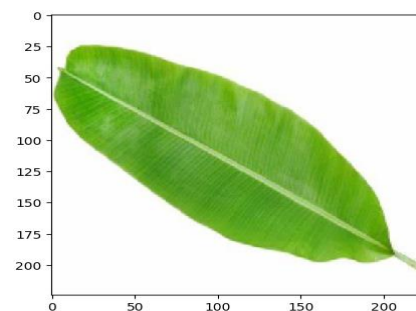
Boron

(b) Boron deficiency



Zinc

(c) Zinc deficiency



Healthy

(d) Healthy leaf

Figure 4.20: Image prediction using leaf datasets

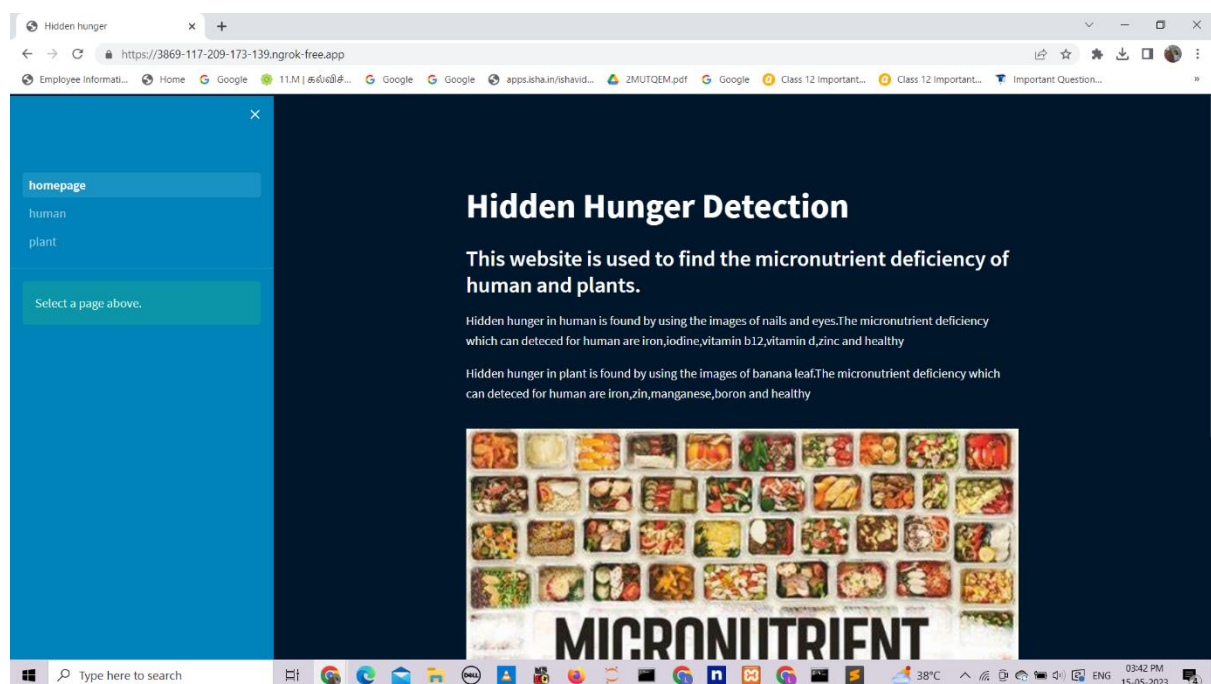
The above figure 4.19 shows the outcomes that are predicted as Manganese, Boron, Zinc deficiency and Healthy leaf using cnn model in leaf dataset. Similarly the deficiencies are predicted using other models such as Inception V3, Xception, MobileNet V2, ResNet 152 V2.

4.4 WEB DEVELOPMENT

Website has been developed for predicting the micronutrient deficiencies with the built model. The streamlit model is used as library for deploying our model to the website. Streamlit is a free and open-source framework to rapidly build and share beautiful machine learning and data science web apps. It is a Python-based library specifically designed for deploying machine learning models into web apps. It runs on the local host. To make this website as a public url, we used ngrok which makes our host as a server and generates url which can be accessed by everyone.

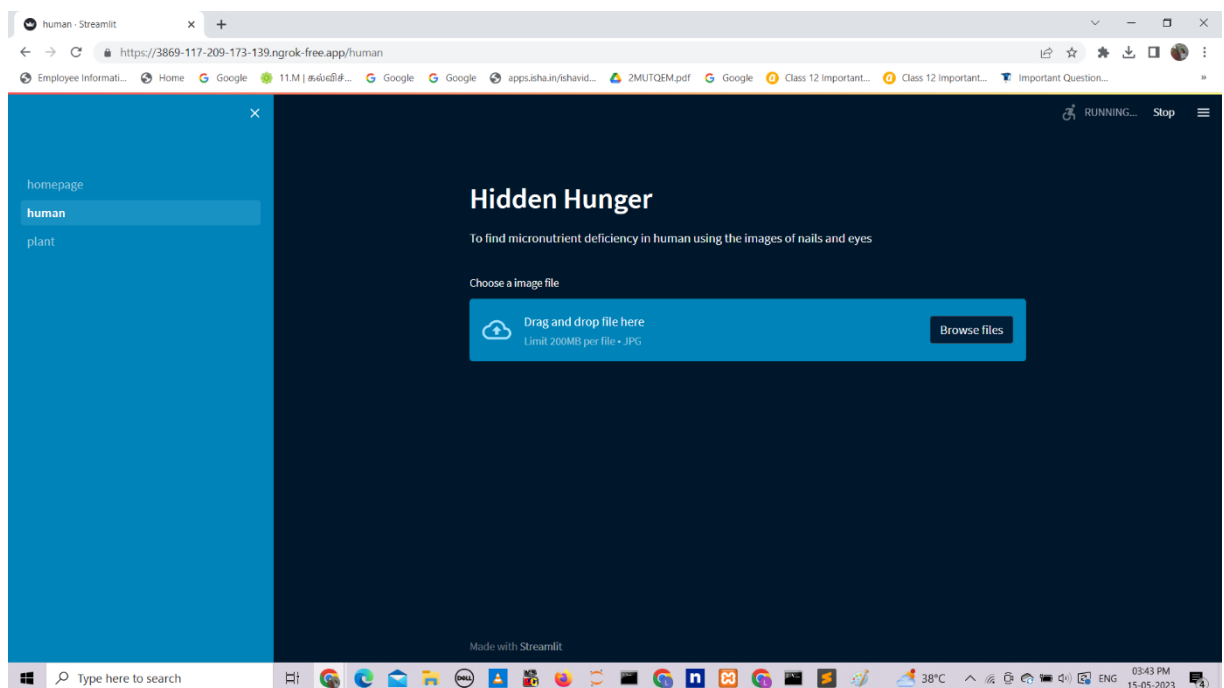
4.4.1. Steps to access the webpage

Step 1: Click on this link <https://3869-117-209-173-139.ngrok-free.app/> to get into our website.

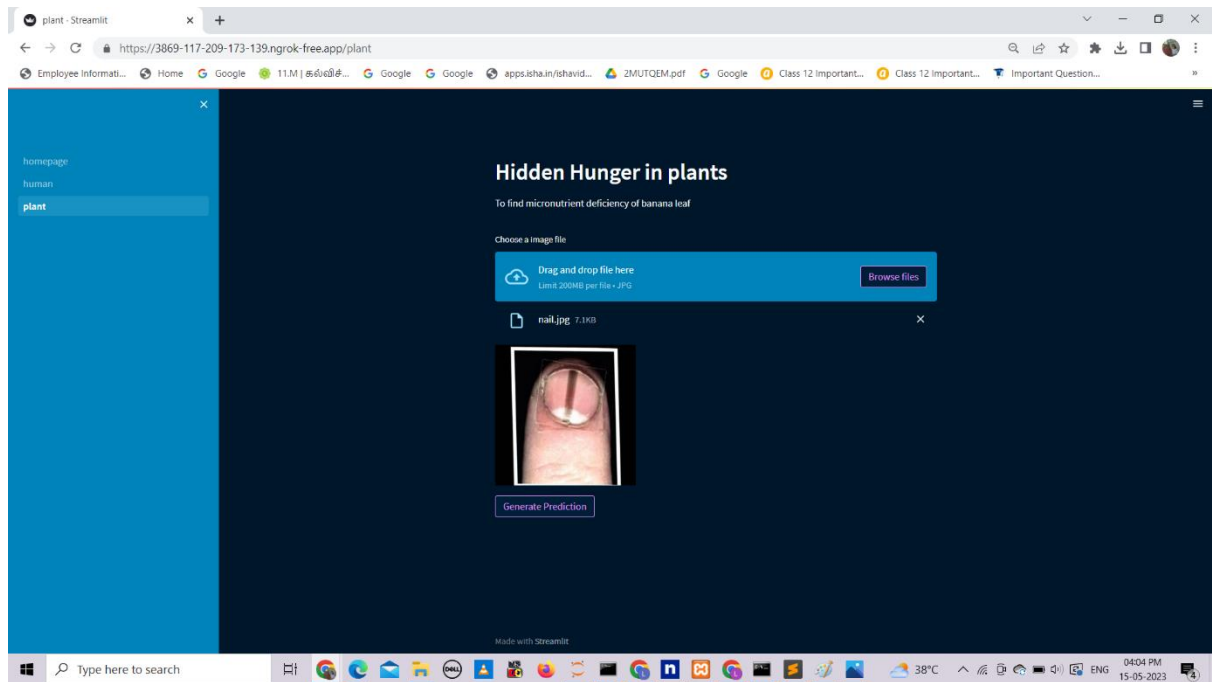


The above figure shows the outlook of our web page. It consists of drop down list that has two options for selecting hidden hunger detection system in human and plant

Step 2: Select human or plant option from the dropdown list depending on where the micronutrient deficiency need to be detected

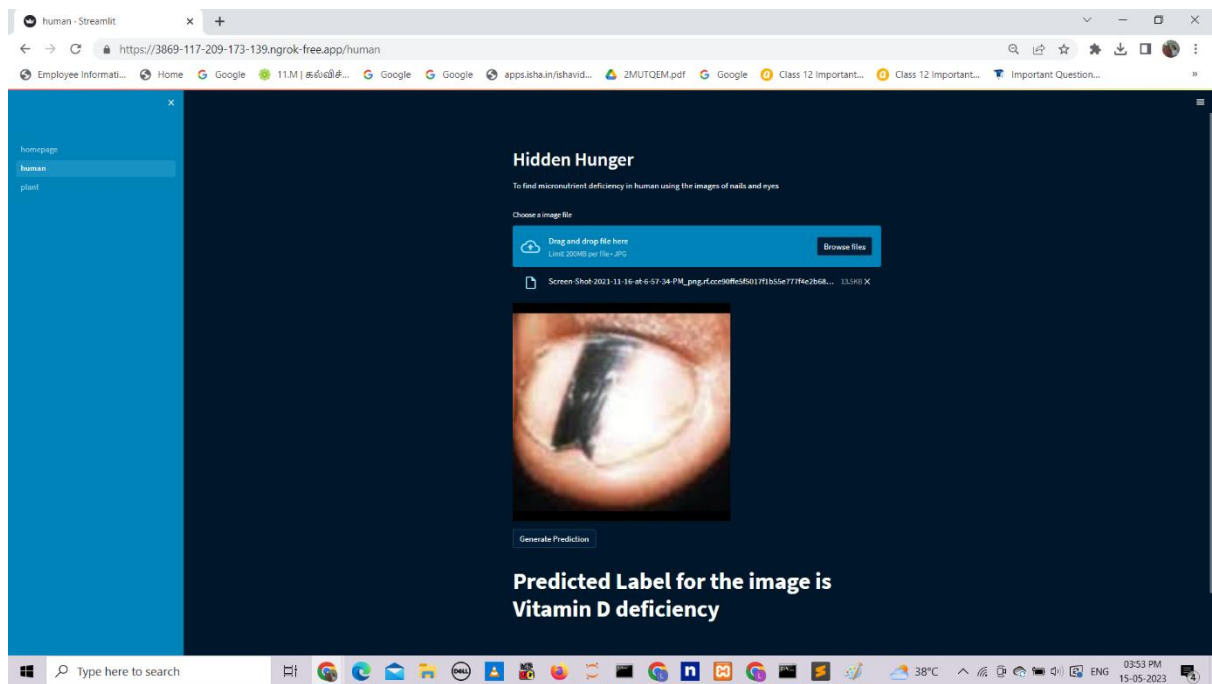


STEP 3: Click on the Browse files option and select the image which needs to be detected

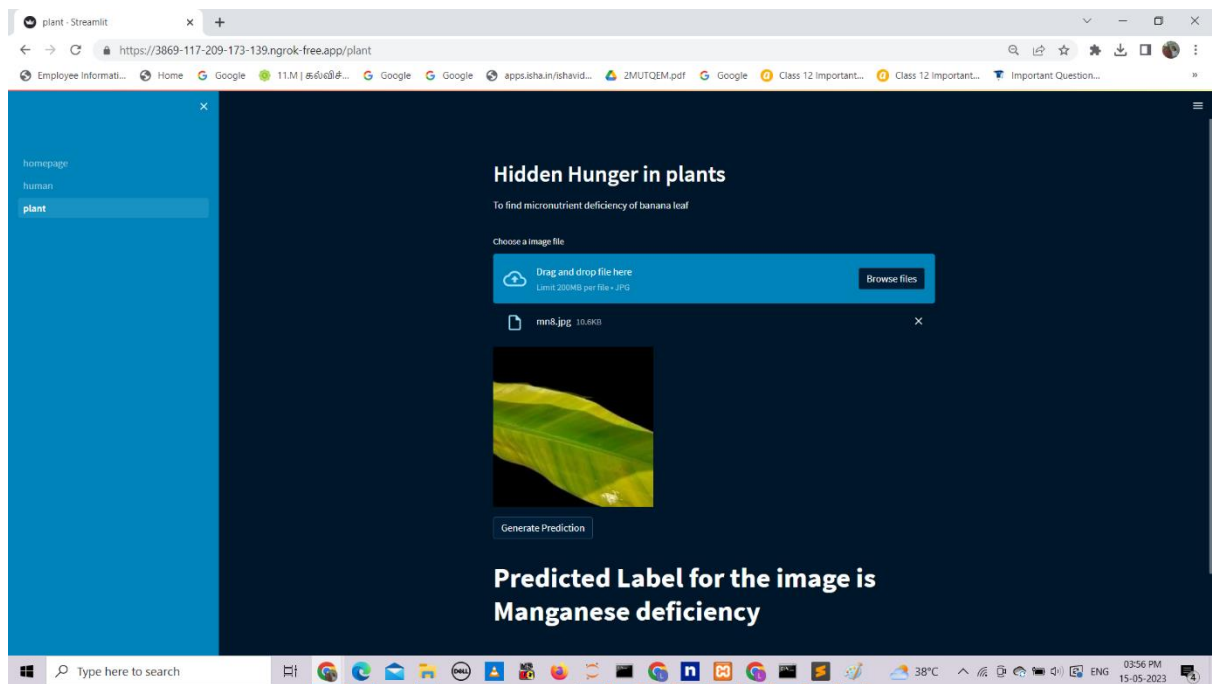


STEP 4: After the selected input image is loaded on to the web page, Click on the Generate prediction option and wait until the prediction is generated.

STEP 5: Finally the predicted result will be displayed



Similarly, the output is predicted for the leaf dataset



CHAPTER 5

SUMMARY AND CONCLUSION

5.1 SUMMARY

The proposed method of detection of micronutrient deficiency in human and plants is performing well in deep learning models like ResNet 152 V2, MobileNet, Inception V3, Xception, Traditional CNN method. Those models were explained along with their architecture in chapter 3. The input dataset is obtained from kaggle and it is augmented, split into train and test for human dataset in the ratio of 8:2 and split into test, train, validation sets for leaf dataset in the ratio of 7:2:1. The number of epochs used here is 20, batch size is 32 and learning rate is of 0.001. The optimizer used here is Adam optimizer. Performance measurements are used to determine the effectiveness of image processing technique in achieving expected results. Among all those models mentioned above ResNet 152 V2 gave the highest validation accuracy of 98% for leaf dataset and 88% for nail dataset. Therefore in order to obtain better accuracy we can use ResNet 152 V2 model.

5.2 CONCLUSION AND FUTURE SCOPE

Our proposed work is to provide an automated and reliable economic solution for hidden hunger identification. As there is no awareness about micro nutrient deficiency among people and also many research works are done on macronutrient deficiency. We had done our project to detect micronutrient deficiency in easier way. The human brain consists of several layers in order to recognize an image or object by detecting its edge, brightness, shape, size etc. Likewise CNN also consists of several layers that are used for image recognition thereby detecting the micronutrient deficiency. For detecting micro nutrient deficiency in human and plants we use healthy and nutrition deficient nail, eye and leaf images. This dataset is given to supervised machine learning as training dataset for further detection and

identification of exact nutrient deficiency in crops and human in order to take preventive measures to attain zero hunger. ResNet model gave the highest validation accuracy and predicted well. The proposed model will be highly useful for detecting the micro nutrient deficiency detection. Future work may be even more increasing the accuracy of the models with more micro nutrient deficiencies. An android app or website may be created for hidden hunger detection system which allows public an easy access to this system.

CHAPTER 6

REFERENCES

1. A. S. Eldeen, M. AitGacem, S. Alghlayini, W. Shehieb and M. Mir, "Vitamin Deficiency Detection Using Image Processing and Neural Network," 2020 Advances in Science and Engineering Technology International Conferences (ASET), Dubai, United Arab Emirates, 2020, pp. 1-5, doi: 10.1109/ASET48392.2020.9118303.
2. Banumathi, J., et al. "An intelligent deep learning based xception model for hyperspectral image analysis and classification." *Computers, Materials & Continua* 67.2 (2021): 2393-2407.
3. Debnath, S., Paul, M., Rahaman, D.M., Debnath, T., Zheng, L., Baby, T., Schmidtke, L.M. and Rogiers, S.Y., 2021. Identifying individual nutrient deficiencies of grapevine leaves using hyperspectral imaging. *Remote Sensing*, 13(16), p.3317.
4. García, Borja & Malounas, Ioannis & Mylonas, Nikolaos & Kasimati, Aikaterini & Fountas, Spyros. (2022). Using EfficientNet and Transfer Learning for Image-Based Diagnosis of Nutrient Deficiencies. *Computers and Electronics in Agriculture*. 196. 10.1016/j.compag.2022.106868Get.
5. <https://github.com/nachi-hebbar/Gradip-FlowerClassification-WebApp>
6. H. Yalcin and S. Razavi, "Plant classification using convolutional neural networks," 2016 *Fifth International Conference on Agro-Geoinformatics (Agro-Geoinformatics)*, Tianjin, China, 2016, pp. 1-5, doi: 10.1109/Agro-Geoinformatics.2016.7577698
7. Jae-Won, C., Trung, T.T., Thien, T.L.H., Geon-Soo, P., Van Dang, C. and Jong-Wook, K., 2018, November. A nutrient deficiency prediction method using deep learning on development of tomato fruits. In 2018 International conference on fuzzy theory and its applications (iFUZZY) (pp. 338-341). IEEE
8. J. Kolli, D. M. Vamsi and V. M. Manikandan, "Plant Disease Detection using Convolutional Neural Network," 2021 IEEE Bombay Section Signature Conference (IBSSC), Gwalior, India, 2021, pp. 1-6, doi: 10.1109/IBSSC53889.2021.9673493.
9. Jose, A., Nandagopalan, S., Ubalanka, V. and Viswanath, D., 2021, May. Detection and classification of nutrient deficiencies in plants using machine learning. In *Journal of Physics: Conference Series* (Vol. 1850, No. 1, p. 012050). IOP Publishing.
10. N. Jmour, S. Zayen and A. Abdelkrim, "Convolutional neural networks for image classification," 2018 International Conference on Advanced Systems and Electric

- Technologies (IC_ASET), Hammamet, Tunisia, 2018, pp. 397-402, doi: 10.1109/ASET.2018.8379889.
11. Rabano, Stephenn L., et al. "Common garbage classification using mobilenet." 2018 IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM). IEEE, 2018.
 12. Sabri, N., Kassim, N.S., Ibrahim, S., Roslan, R., Mangshor, N.N.A. and Ibrahim, Z., 2020. Nutrient deficiency detection in maize (*Zea mays* L.) leaves using image processing. *IAES International Journal of Artificial Intelligence*, 9(2), p.304.
 13. Sharma, M., Nath, K., Sharma, R.K., Kumar, C.J. and Chaudhary, A., 2022. Ensemble averaging of transfer learning models for identification of nutritional deficiency in rice plant. *Electronics*, 11(1), p.148
 14. Taha, M.F., Abdalla, A., ElMasry, G., Gouda, M., Zhou, L., Zhao, N., Liang, N., Niu, Z., Hassanein, A., Al-Rejaie, S. and He, Y., 2022. Using deep convolutional neural network for image-based diagnosis of nutrient deficiencies in plants grown in aquaponics. *Chemosensors*, 10(2), p.45
 15. Tran, T.T., Choi, J.W., Le, T.T.H. and Kim, J.W., 2019. A comparative study of deep CNN in forecasting and classifying the macronutrient deficiencies on development of tomato plant. *Applied Sciences*, 9(8), p.1601
 16. Tripathi, Milan. "Analysis of convolutional neural network based image classification techniques." *Journal of Innovative Image Processing (JIIP)* 3.02 (2021): 100-117.
 17. Waheed, H., Zafar, N., Akram, W., Manzoor, A., Gani, A. and Islam, S.U., 2022. Deep Learning Based Disease, Pest Pattern and Nutritional Deficiency Detection System for "Zingiberaceae" Crop. *Agriculture*, 12(6), p.742
 18. Watchareeruetai, Ukrit & Noinongyao, Pavit & Wattanapaiboonsuk, Chaiwat & Khantiviriya, Puriwat & Duangsrissai, Sutsawat. (2018). Identification of Plant Nutrient Deficiencies Using Convolutional Neural Networks. 1-4. 10.1109/IEECON.2018.8712217.
 19. Wulandhari, L.A., Gunawan, A.A.S., Qurania, A., Harsani, P., Tarawan, T.F. and Hermawan, R.F., 2019. Plant nutrient deficiency detection using deep convolutional neural network. *ICIC Express Lett*, 13(10), pp.971-977
 20. Wulandhari, Lili Ayu et al. "PLANT NUTRIENT DEFICIENCY DETECTION USING DEEP CONVOLUTIONAL NEURAL NETWORK." (2019).