Identifying Malnutrition from Facial Features of Children Using Deep Learning

by

Nahin Hossain
24341075
Saiyeda Sabiha
23241133
Ahnaf Bin Obaid
20201044
Nawroze Baizid Mohammed
24341076
Mohammad Farhan Labib Al-kafi
20201023

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
October 2024

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.

2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.

3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.

4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Nahin Hossain
24341075

Saiyeda Sabiha
23241133

Ahnaf Bin Obaid
20201044

Nawroze Baizid Mohammed
24341076

Mohammad Farhan Labib Al-kafi
20201023

Approval

The thesis/project titled "Identifying Malnutrition from Facial Features of Children Using Deep Learning" submitted by

1. Nahin Hossain(24341075)
2. Saiyeda Sabiha(23241133)
3. Ahnaf Bin Obaid(20201044)
4. Nawroze Baizid Mohammed(24341076)
5. Mohammad Farhan Labib Al-kafi(20201023)

Of Summer, 2024 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on October, 2024.

Examining Committee:

Supervisor:
(Member)

Dr. Md. Ashraful Alam
Associate Professor
Department of Computer Science and Engineering
Brac University

Program Coordinator:
(Member)

Md. Golam Rabiul Alam, PhD
Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi, PhD
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

Malnutrition remains a significant global health issue, particularly affecting children, leading to severe developmental challenges. Current diagnostic methods are often inefficient, relying on time-consuming physical assessments. This study explores the potential of deep learning techniques to automate the detection of malnutrition in children by analyzing facial images. Several state-of-the-art Convolutional Neural Networks (CNNs), including VGG16, VGG19, InceptionV3, MobileNetV2, DenseNet121, and a custom MobileNetV2, were trained and evaluated on a dataset of 1,953 images. After augmentation, the dataset increased to 9,765 images to enhance the models' performance. Among these models, the custom MobileNetV2 achieved the highest accuracy, with 99.23% during training and 98.57% on validation data, significantly outperforming other architectures. This demonstrates the effectiveness of using CNNs for malnutrition detection. The results highlight the potential of AI-driven solutions to offer faster and more accurate assessments, which could be particularly beneficial in resource-constrained settings. Future research will focus on improving model robustness by incorporating a more diverse dataset, additional health indicators such as body mass index (BMI), and further optimization for mobile application deployment. These advancements could help healthcare professionals to detect malnutrition early and provide timely interventions to improve child health outcomes.

Keywords: Malnutrition, Deep Learning, InceptionV3, VGG19, VGG16, DenseNet121, MobileNetV2

Acknowledgement

We are grateful to God for allowing us to finish our thesis without any significant setbacks. Second, thanks to Mr. Dr. Md. Ashraful Alam, our supervisor, for his thoughtful counsel and support throughout our effort. He was always there to support us. It also may not have been possible without the support of our parents during our journey. With your kind prayers and support, we are almost done with our degree.

Table of Contents

List of Figures

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

AUC Area Under The Curve

BMI Body Mass Index

CNN Convolutional Neural Network

KNN K-Nearest Neighbors

$LSTM$ Long Short-Term Memory Networks

MAE Mean Absolute Error

ML Machine Learning

$MTCNN$ Multi-Task Convolutional Neural Network

RF Random Forest

$RMSE$ Root Mean Square Error

Chapter 1

Introduction

Malnutrition is a huge issue both for rich and poor countries [9] and could even be worsening. The most common cause of malnutrition is lack of consumption of the right amount of nutrients and energy, that leads to health issues such as stunting, wasting and micronutrient deficiencies mostly for children. About two-thirds (approximately) of children less than five years in age are malnourished in India [1], 5-8% being severely malnourished and the rest having mild to moderate forms of malnutrition. The high prevalence of malnourishment (based on a study [2] of 720 children in Bangladesh (Jamalpur)). In particular the number of children underweight was 34.3 percent, stunted at 41.5 percent and the wasting rate at 18.9 percent. The fact this pervades malnutrition in child health, means it is one of the common illnesses affecting them.

There are limitations and drawbacks of traditional approaches to the detection of malnutrition. Our more efficient solution to this challenge is enabled by the massive rapid developments in deep learning and computer vision [8]. Artificial intelligence (AI) has made great progress in using deep learning in a variety of visual tasks such as object detection, image categorization, and facial recognition. Researchers and health experts have also started to examine how facial characteristics could be used, through deep learning algorithms, to spot signs of malnutrition in children. Face recognition is a subset of AI that involves algorithms that automatically analyze facial features like a person's eyes, nose and mouth in order to identify who an image represents. This technology not just helps in health science but also helps doctors, nurses, physicians to identify the malnourished children by their facial traits.

Artificial intelligence in general, and the deep learning within that, can be characterized as a specialized area with focused training of complicated artificial neural networks – this time, with many layers. This subfield deals with automatic extraction of fine features and representations in data. Deep learning makes its way distinct from other artificial intelligence strategies in that it incorporates many layers, which are called deep architectures, inside these neural networks. This multi-layer structure allows the networks to train and learn discriminating hierarchical features and representations from input data autonomously, reflecting its 'deep' learning taste.

In this paper, we intend to tackle the problem of malnutrition of children using the capacities of advanced neural network architectures such as the VGG16, VGG19, InceptionV2, MobileNetV2 and DenseNet121 models. These models have established themselves as leaders in the field of image recognition and classification, and now present us with a chance to explore them in the direction of detecting malnutrition with basic facial analysis.

1.1 Problem Statement

In the field of health diagnostics, deep learning techniques learn to solve the critical problem of child malnutrition through analyzing body and facial features. The objective of this study is to build a descriptive deep learning model for identifying malnutrition symptoms using the images of children. In this focus, we will be dealing with advanced deep learning algorithms, generally using CNN architectures, such as VGG19, VGG16, InceptionV3, DenseNet121, and MobileNetV2 which are successfully used in image classification and feature extraction. The models will be trained rigorously using a vast dataset of images from malnourished and well nourished children, in order to distinguish the two. The automated method described here would significantly reduce the time needed for malnutrition detection, bringing healthcare providers a fast and reliable way to identify at risk children and avoid negative outcomes.

1.2 Research Objective

The idea is to develop an intelligent solution that will help to diagnose cases of malnutrition in children based on their facial patterns. Doctors and nurses will be able to diagnose malnutrition in kids on time, enabling them to provide proper treatment for the children early on. The research objectives for this paper are:

1.3 Report Organization

Our report consists of five chapters, each covering different aspects of the project.

Chapter 1 provides an overview, including sections such as 1.1 Problem Statement, 1.2 Research Objectives, and 1.3 Report Organization.

In Chapter 2, we discussed about the Related Works.

The research methodology, along with its subsections, is presented in Chapter 3, which includes 3.1 Data Collection, 3.2 Data Augmentation, 3.3 Data Preprocessing, 3.4 Model Selection and 3.5 Proposed Model.

In Chapter 4, we discuss about the experimental result, including 4.1 Results Analysis, 4.2 Model Evaluation and 4.3 Discussion.

In Chapter 5, we delve into the discussion with sections like 5.1 Conclusions, and 5.2 Future Works

Throughout each section, we have created scenarios to support and enhance our research.

1. To investigate the applicability of deep learning models for detecting malnutrition in children through facial feature analysis.

2. To evaluate and compare the performance of various convolutional neural network (CNN) architectures, including VGG16, VGG19, InceptionV3, MobileNetV2, DenseNet121, and a custom MobileNetV2, for malnutrition detection.

3. To develop an automated malnutrition detection system based on facial image classification that can assist healthcare professionals in early identification of malnutrition in children.

4. To optimize the selected models through techniques such as transfer learning and data augmentation to improve the accuracy and robustness of the malnutrition detection system.

Chapter 2

Literature Review

Using the power of AI, the authors in [4], investigated the potential relationships that exist between facial features seen in photos and their respective BMI. While trying to find out if BMI can be determined through facial photos, they concluded that facial features from photos can be used as a form of biometric identification. They proposed a deep residual neural network method, ResNet, to derive the BMI using facial photos. MTCNN is used for their model, with which they can locate facial features with a higher degree of accuracy and thereby increase the accuracy of face detection. Experiments were carried out on 1,530-record datasets. Ample data preprocessing was carried out to handle missing values, convert height, and ultimately compute BMI. They also take into consideration performances via metrics: AUC, MAE, and RMSE. Comparisons against previous methods give the impression that this technology might just be a game-changer in health assessments through analyzing images online. The results are reported to clinicians along with related pictures and features. The details of the workflow for this project are given using an explicit flowchart that is needed for understanding and application.

In [5], investigators studied the influence of nutrition on child health in underprivileged areas of Ethiopia. They analyzed data taken from USAID and also used CIAF-Composite Index for Anthropometric Failure to identify malnutrition. However, these traditional approaches were not efficient in handling multi-variables. Contrarily, machine learning was a more accurate method to that end. It was then divided for training and testing/optimization, where several algorithms were applied, but RF proved to be the best. The results of the analysis of more than 29,000 children are that almost half of them are suffering from malnutrition, and the potential significant factors are parental education, rural residence, poverty, and poor access to media and sanitation. The RF model far outperformed traditional techniques by a huge margin, hence proving the potential of machine learning in child malnutrition predictions. These findings have shown areas where interventions are most needed. To the policymakers, this is a focused approach towards solving the problem. Future research should, therefore, focus on further refining these models for a holistic approach toward the health of the children.

While studying the specifics of malnutrition, thus it is crucial to use accurate prediction methods. Analyzing [8], the base TabNet and Random Forest H2O are used to identify Risk factors for malnutrition reflected in IDHS 2005-2006 and 2015-2016 datasets. Maternal, child, and household information data sets in the form of ”Children Recode” files, derived from these surveys, form the analytical bedrock. Using the data collected between the years 2015-16, TabNet reports the highest overall accuracy (74.53%) and the AUC- ROC score of (82.55%) for Stunting (HA index). Being accurate at Wasting (WH index), TabNet achieves 86.30% as compared to the Random Forest set in H2O with AUC-ROC scores of 99.95%. For the concurrent Stunted-Wasting index, Maintaining the same trends, TabNet reaches the apex of accuracy 96.46% and the second-best AUC-ROC is delivered by the Random Forest model in H2O with 98.53%. The results show the high effectiveness of both models for all significant nutritional parameters. It could also be a direction for future research to fine tune these models with a view to getting a more rounded picture of malnutrition.

The study [12] focuses on one of the health threats facing the society that of malnutrition during COVID-19 pandemic with the introduction of a smartphone approach to raising height, weight, And different important health indicators that are otherwise so easily accessible. This makes the use of this innovative system employ 3D, facial, body and gesture recognition in record times and meta-characters, performing after shooting the subject full-body pictures with the device’s camera, preceded by thorough image analysis. The methodology involves some initial steps such as face and full body detection and recovery of 3D models and features which are concluded into an Android application prototype. Terse highlights show that high-precision 3D depiction, multimodal learning, and edge device utilization is effective for high grades of state of the art weight predictions. This shows that benchmark tests support this new model over previous methods. As directions for further research, the study is intended to further improve the approaches regarding training and orientation and to establish a higher reliability of height prognosis. The role of the research is underlined by the use of data received from 30 volunteers and owing to the availability of Dr. Min Jiang’s data set. In a COVID affected environment, the paper provides the first glimpse of a mobile first approach to combat malnutrition and promote health surveillance.

The study [14] tackles the critical issue of child malnutrition, particularly in children under five, a condition that is widespread in developing countries. The study introduces a new method that utilizes data mining and machine learning techniques in order to forecast malnutrition depending on key factors such as height-for-age (HAZ), gender, weight-for-age (WAZ) and age. Two core categorization techniques, the Bayesian classifier and K-nearest neighbor (KNN), are explored. Conventional methods of data analysis are often slow with lower accuracy; thus, machine learning offers a more efficient and precise alternative. The study also explores the use of other classification techniques, such as Decision Trees, to classify children into different nutritional statuses, including stunted growth, underweight, and wasting. The proposed system aims to develop a real-time application using Visual Studio and SQL Server, offering timely insights for healthcare providers to facilitate swift interventions. Data is gathered from platforms such as Kaggle and Dataworld. Though the Naive Bayes and KNN algorithms are central to the system, the paper also elaborates on the steps and functionalities involved. The ultimate aim is to support healthcare practitioners by offering predictive capabilities and dietary guidance to address the significant issue of child malnutrition.

The study [6] presents a novel approach to the automatic detection of malnutrition in children under five years old using Convolutional Neural Networks (CNNs), mainly leveraging the AlexNet model.The system uses Transfer Learning to analyze images and determine children's nutritional status, achieving 96% accuracy with a 0.001 learning rate during evaluation. The system classifies children into three categories—Malnourished, At Risk, and Normal—using image analysis and z-scores for Weight-for-Age (WAZ), Weight-for-Height (WHZ), and Height-for-Age (HAZ). Malnutrition continues to be a serious global issue, with WHO data from 2020 indicating that 5.6% are overweight, 6.9% are wasted and 21.3% of children under five are stunted. Findings from the National Family Health Survey in India show continuing battle with malnutrition. Also, other methods were studied such as color data analysis, improvements in anthropometric measurements, Clinical Nutrition Specialists (CNS) work, and alternative predictive models. Further, research should seek to make CNNs better able to differentiate between malnutrition types. The intent of this approach is to offer deeper lessons for combating malnutrition and strengthening more effective interventions.

The architectures of the convolutional neural networks VGG19 and ResNet50 are described in the work [7]. The VGG19 is equipped with 16 convolution layers, 5 levels of max-pooling, three fully connected layers, as well as 1 softmax layer. This design has a filter size of 64, 128, and 256 during the convolutional layer. This analysis gave many details about this VGG19 and especially about this amazing dimension, telling us that several layers are useful to detect these features. While the paper considered ResNet50, another type of convolutional neural network with 48 convolutions, 1 maximum pooling, and 1 average pooling. A new feature of ResNet50 is that using fast connections makes it possible to implement the functions of residues. Dial-ar knew that shortcut connections gave the choice to jump over some of the tiers of the network. Shortcut connections aid in reducing retraining error in five deep neural networks with several stacks of layers created. Thus, indicating in the analysis carried out on the study, how it elaborates at an excellent level, the weaknesses of the designs of neural networks with several layers with the architecture of ResNet50. In general, ResNet50 had better performance in training errors through the implementation of shortcut links and residual functions, hence promoting enhancement of training precision as well as future outlooks in real image classification. In the presentation, the author explained how ResNet50 was created to solve deep neural network problems and to show how it could be helpful in utilizing applications instead of the conventional design, VGG19, for analytical image classification.

The study [3] is therefore dedicated to proposing an image classification model utilizing MobileNetV2, a modern lightweight deep learning model intended for mobile and other embedded vision applications. It has its root from the MobileNetV1 architecture with added techniques of depth wise separated convolution, linear bottleneck and inverted residual connection. These innovations are essential in attempting to solve problems of high computational requirements while still achieving high image classification accuracy. I particularly manage to capture how the study avoids the use of numerous parameters by using depthwise separable convolutions in order to make it faster and suitable for use in mobile. Also, MobileNetV2 enhances on inverted residuals and linear bottlenecks, it addresses the issue of inhibiting features by ensuring that information flow does not undergo a thorough transformation during a convolution process. Such enhancements in architectural designs are first run and evaluated on datasets by adopting criteria such as test set accuracy and training time. Analyzing these methods, the paper underlines that MobileNetV2 can provide the high rate of successful classification of images with conditions of required low computational resources. Such capabilities make the algorithm especially useful in real-time applications where computation capabilities of the devices are a limiting factor on the IoT frontier such as smartphones and IoT devices. Using T-SNE for visualization and dimensionality reduction to further illustrate how the model effectively inspects large scale image datasets, the paper elaborates more skills in this aspect.

In this following study [11] we get to see that the identification of facial paralysis for the diagnosis of stroke has been enhanced through the recent deep learning models in the form of DensityNet121, a convolutional neural network that specializes in biomedical pictures. This model is especially successful in assessing facial weakness due to strokes by analyzing such factors as eye movement, imbalance of the mouth, and wrinkles. The study employs static face photos enhanced by employing data augmentation means like cropping, flipping, and resizing to enhance the model capacity despite limited data. Compared with the other researched networks, DenseNet121, with a highly connected structure, can well develop relevant characteristics for face features, enhancing accuracy. This Kaggle dataset contains both normal and afflicted people, and that makes it have the primary diversity that is used to train the model. Standard measures of performance like accuracy and computation capacity reveal the capability of differentiating the stroke patients by using small distinguishing tendencies in facial features. Not only does this technology hasten the diagnosis of the stroke, but it offers an adjunct or less invasive means over the current methods with which stroke can be diagnosed and hence be treated early, thereby making this technology an essential enabler.

In this study [13] they have chosen FER as one of the main areas of focus because the CNN used in this work demonstrated high accuracy in emotion recognition. For standard CNNs, the severity identification accuracy rates of such algorithms have exceeded 70%, and with more advanced techniques involving the transfer learning, the best percentage achieved was 92.42%. This feature of parallel convolutional filters clearly deciphers this recent CNN model known as InceptionV3, as it helps to capture both local as well as global facial neuroscience features, leading to a successful prevalence over previous showpiece network models. With the data sets such as the Karolinska Directed Emotional Faces (KDEF) that embrace a wide range of emotions and through processes such as image normalization and converting the pictures to grayscale, it has been realized that the score of our Inception V3 models has reached a 91.33% accuracy. Nonetheless, improvement was extended by joining CNNs with LSTM classifiers, and some approaches reached the accuracy level beyond 99%. The evolutions with respect to the CNN design and data processing algorithms ensure that the FER will continue to deliver improvements to areas of health, learning, and human-computer interfaces.

In the study [10] the literature on facial emotion detection shows a range of strategies and methods used, as well as differing research approaches, and that indicates the development of approaches used in facial emotion recognition based on deep learning. Pandey [1] in his work used a basic structure of CNN that comprises of five convolutional layers, which has 76.62% validation accuracy using the minimally preprocessed FER2013 dataset, and Dar [2] combined some changes to the Efficient-SwishNet model with 100% accuracy on the CK+ dataset. Pham [3] provided an innovative architecture known as the Residual Masking Network and produced diverse accuracy levels with different datasets, implying that architecture design plays a critical role in performance results. The CoAtNet architecture described by Dai [8] has been designed to merge CNNs and vision transformers (ViTs) and has demonstrated near-optimal performance with limited resources. Moreover, the paper conducted by Hardjadinata [10] compared the facial expression recognition on Xception and DenseNet architectures, indicating that there are different choices of models for emotion recognition. Last, multimodal techniques highlighted by Agarwal [14] also focuses on the fact that using facial landmark data improves image processing, outlining the general tendency towards increasing the accuracy due to data fusion. Together, these papers point toward an arc of increasing complexity of models rendered through the use of mixed hardware-software systems and multiple data modalities toward the creation of facial emotion detection algorithms that are more robust and accurate.

In the study [10], the hybrid model of CoAtNet - a combination of Vision Transformers (ViTs) and Convolutional Neural Networks (CNNs) – is used to develop a facial emotion detection system. First, the system is trained and validated by the FER2013 dataset, a popular benchmark in emotion recognition applications. The models are compared to different popular CNN models, VGG16, DenseNet121, ResNet50 and Xception, and it is found that the CoAtNet model reaches the highest validation accuracy of 80.314%. Furthermore, the paper demonstrates an ingenious method for increasing the performance of these deep learning models with facial landmark data by using a Multimodal Fusion technique, further increasing accuracy, and ending with a final validation accuracy of 83.458% for CoAtNet.

Chapter 3

Methodology

In this section we are going to elaborate the proposed methodology for our work which includes data collection, data cleaning, preprocessing, model training and evaluation. Diagram of the overall methodology is shown in Figure 3.1 and every step of the methodology is further discussed in the following subsections.

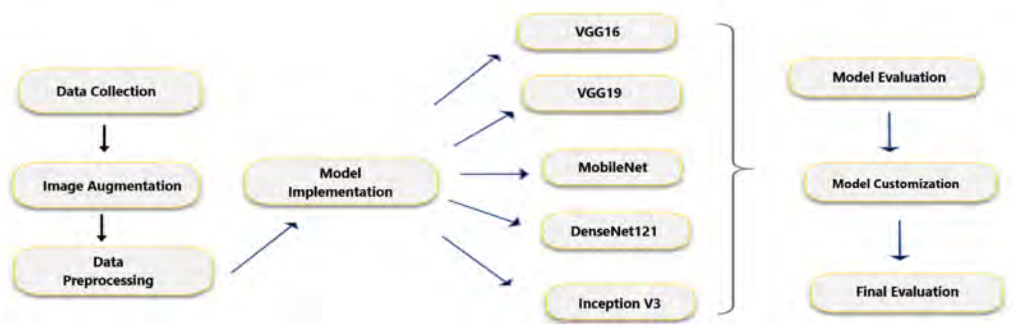

Figure 3.1: Proposed Methodology

3.1 Data Collection

In order to conduct our research, we have used secondary dataset collected from Kaggle. The dataset comprises 1953 photos which divided into 2 categories Nutrition and Malnutrition. The distribution of the images is given below in Table 3.1.

Table 3.1: Distribution of Images by Class

Class	**Number of Images**
Nutrition	1002
Malnutrition	951
Total	1953

3.2 Data Augmentation

Image data augmentation is the process of generating new transformed versions of images from the given image dataset to increase its diversity. To create diversity, we used augmentation techniques. We transform one image into five images. The distribution of the images after applying augmentation is given below in Table 3.2.

Table 3.2: Distribution of Images Before and After Augmentation

Class	**Original Image**	**After Augmentation**
Nutrition	1002	5010
Malnutrition	951	4755
Total	1953	9765

3.3 Data Preprocessing

We carried out image rescale using (1/255) that is aimed at changing the pixels of the images from the (0, 255) range to a range of (0, 1) so that the images do not dominate in case of loss. The modified images have a fixed resolution of 224 by 224 pixels. A total of 9764 images belonging to two categories have been internally and externally stored (Google Drive) for future purposes. All these images are utilized for the purpose of training, testing and validation of the model. In total 7812 images were selected for training the model. In the same procedure of creating training dataset, to prevent the model from being biased, the testing dataset and validation dataset are also created in a different way which separates the training dataset into portions. The testing dataset portion is made up of 976 images and the validation dataset portion consists of 977 images for the model's prediction.

3.4 Model Section

3.4.1 VGG16

There are sixteen VN layers in VGG 16 – the layering is sequential with 13 convolutional and 3 fully connected in the end. All the convolutional layers use 3x3 kernels of unit strides. This is useful in capturing the spatial and finer details within a pixel while at the sametime preventing the rise of countless trainable parameters. Max pooling takes place after some convolutional blocks to aid in the further reduction of the feature maps and softmax classification is done on the remaining 3 fully connected layers. Implementation of VGG16 is not restricted by its complexity and such a simple architecture achieves great results in large scale image classification tasks even without deep learning such as in ImageNet. This is VGG's unique feature, the architecture works on different levels of abstraction hence cannot miss out on any detail in the images. Unfortunately, the storage requirement to make this neural network work is more than 138 million which is quite expensive. The model also suffers from overfitting drawbacks and in most cases with help of such a structure it is impossible to train on smaller datasets. This is caused by the huge inflow of parameters, which also makes the training a long process. The essential analyzed features are the thirteen convolutional layers, 3x3 filters, max pooling layers, and around 138 million parameters.

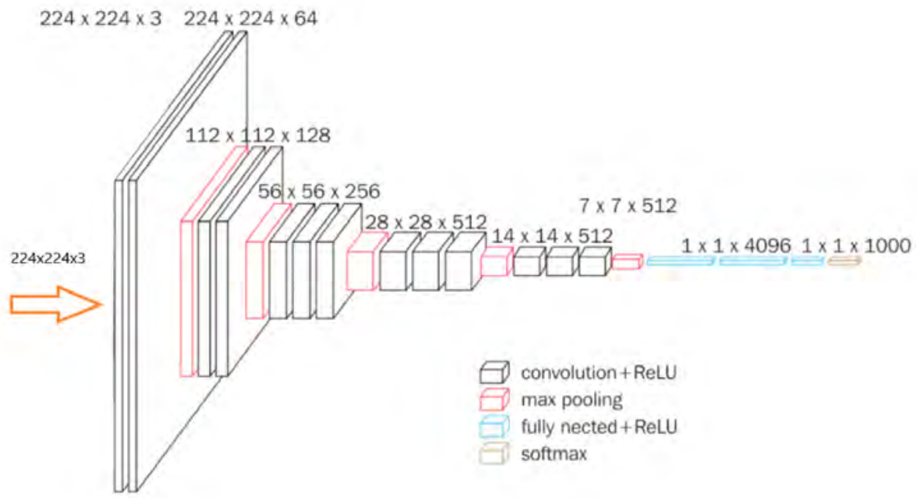

Figure 3.2: Architecture of VGG16

3.4.2 VGG19

Just like VGG16, VGG19 is designed in a layered form but consists of more convolution layers than VGG16 and VGG 19 contains 16 convolutional layers 3 fully connected layers thus totalling to 19 weight layers. As VGG16 it features 3x3 size filters, max pooling, and fully connected layers for the purpose of classification. The extra layers in the VGG19 structure serve the purpose of enabling it learning more complex features and deeper representations making it very suitable for image classification and object recognition on rich datasets. Despite its advantages, VGG19 is much more resource-hungry as it contains 144 traces, and therefore will be slower in training even than VGG16. The wider architecture also poses the challenge of overfitting, especially for small data sets. Essential attributes consist of 3x3 convolutions, max pooling, 16 convolutional layers and billions of parameters rendering it more limited to computational resources than VGG16.

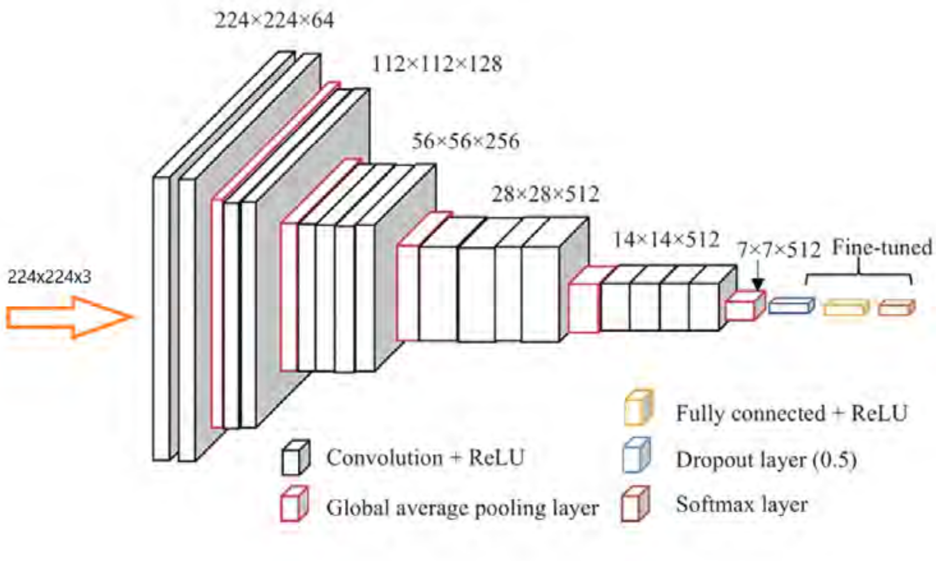

Figure 3.3: Architecture of VGG19

3.4.3 InceptionV3

InceptionV3 is an effective deep learning model which is based on the Inception architecture where images are convoluted, with several filters applied at the same location, i.e., 1x1, 3x3 and 5x5. The results are then combined together so that both the local as well as the global structure in an image can be well understood. It is effective in performing a myriad of tasks due to the multi-scale nature of the feature extraction. InceptionV3 also employs factorized convolutions (splitting big convolutions into smaller pieces), auxiliary classifiers to improve gradient flow and batch normalization to help in speed and stability during training. InceptionV3 has about 23 million numbers of adjustable weight parameters, which makes it less demanding in terms of computing resources than VGG models but able to remain highly precise. However, the existence of several branches makes it harder to make, use and fine-tune and although it is economically designed, training still needs high-end computers. Main features include the Inception module with parallel convolutions, factorized convolutions, auxiliary classifiers, and about 23 million parameters.

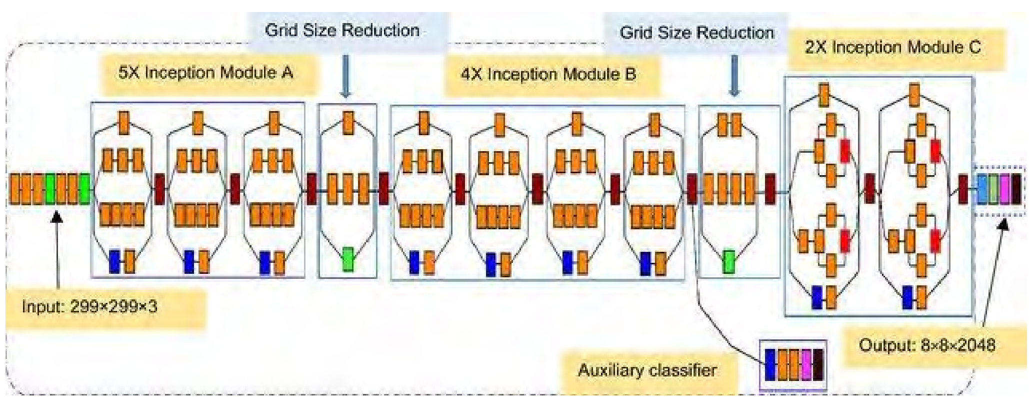

Figure 3.4: Architecture of InceptionV3

3.4.4 MobileNetV2

MobileNetV2 is aimed at devices with limited resources like mobile phones and embedded systems, and it takes as a foundation the concept of depthwise separable convolutions availed in MobileNetV1. Such convolutions divide the process into two distinct stages, namely, depthwise which considers only one channel at a time and pointwise (1x1), thus leading to reduced costs. In MobileNetV2, inverted residual blocks are introduced with the goal to take in data, reduce it and then expand it. To enable these transformations, linear bottlenecks are incorporated to help with information retention. MobileNetV2 has struck a balance between architecture complexity and accuracy with about 3.4 million parameters and fast inference performance making it suitable for real time applications. Small model size comes with a drawback of reduced accuracy, when compared with larger architectures like VGG or Inception, as well as a limited ability to model very complex patterns. Notable elements include depthwise separable convolutions, inverted residual blocks and linear bottlenecks among other elements to enhance its efficiency.

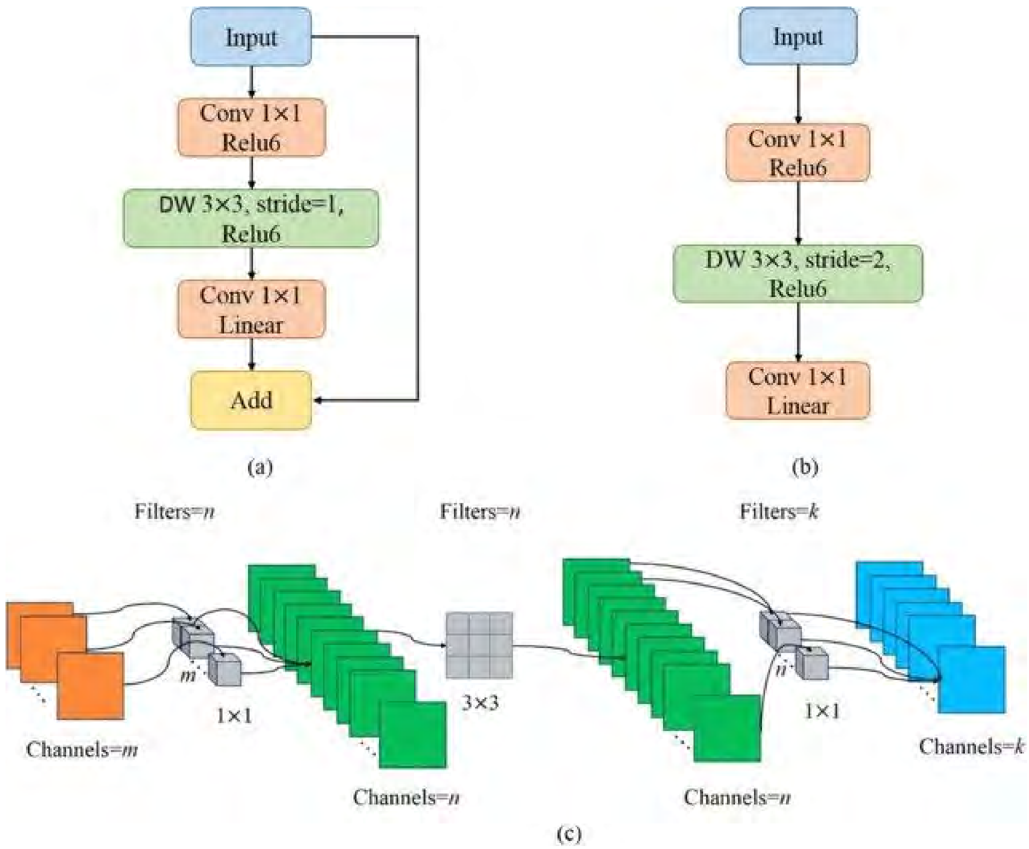

Figure 3.5: Architecture of MobileNetV2

3.4.5 DenseNet121

DenseNet121 introduces pioneer architecture since it directly implements feed-forward connectivity to all the layers even the ones that came before it and this is different from the residual connections used in the ResNet for example. In DenseNet121's case, this helps as one of the design philosophies, feature reuse, Each layer takes as input the outputs of all prior layers which means that this architecture makes very good use of parameters. In this sequence, Dense blocks are followed by transition layers in the architecture which do batch normalization, 1x1 convolutions, and 2x2 average pooling for reducing the size. DenseNet121 has an approximated 8 million parameters, which is less when compared to computationally intensive networks like VGG, and dense connections assist in improving gradient flow thus lessening the effect of fading gradients and enabling training of deeper networks. Nevertheless, the drawback of the dense connectivity is that it results in high memory consumption during training attributed to the feature maps and its implementation is challenging due to the intricacies of its design. Densely connected layers, downsampling transition layers, and low parameter cost are the main characteristics.

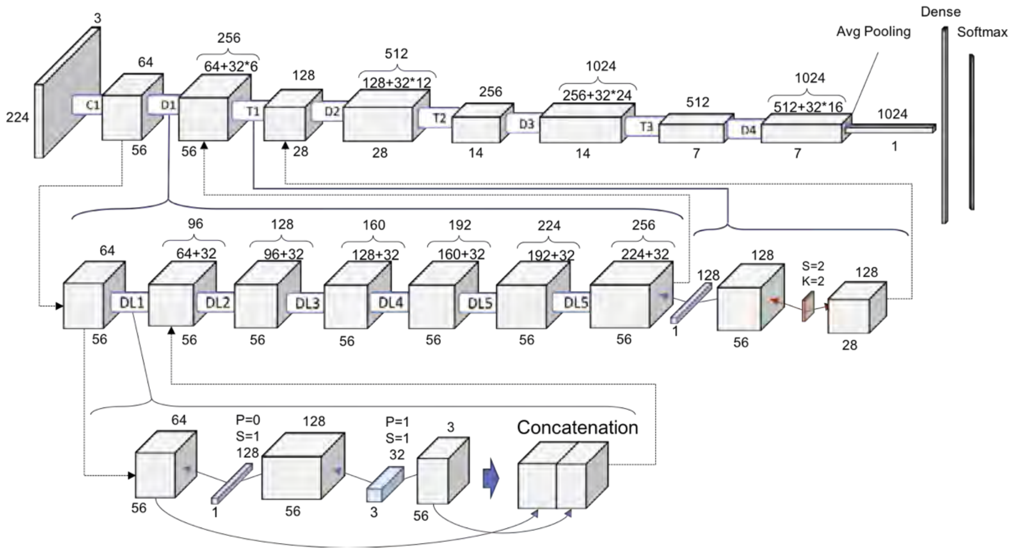

Figure 3.6: Architecture of DenseNet121

3.5 Proposed Model

3.5.1 Custom MobileNetV2

Initial Stage (First 10 Epochs):

To begin with, a model is created based on the MobileNetV2 architecture with pre-trained weights on ImageNet. The top classification layer is omitted ('include_top=False') and remaining layers are kept frozen ('trainable=False') so that their weights will not be modified in this stage. The images fed into the network are resized to 224×224 pixels in order to suit the input size requirements of the MobileNetV2 model. After the base model a GlobalAveragePooling2D layer is placed, which serves for reducing the extracted feature maps dimensionality before the final Dense layer with softmax activation which classifies the output to the number of classes needed ('class_count'). The model was additionally optimized according to the Adam optimizer with 0.0001 learning rate, whereas the loss function was categorical cross-entropy. Within these 10 epochs, the frozen base model provides the existing network with already learned features while allowing only new top layers added to the model for classification to be trained, this ensures a good understanding of the present architecture before moving to the next stage.

Customization Stage (Next 10 Epochs):

When the initial 10 epochs have been completed, the model shifts to the next stage. In this stage, the base MobileNetV2 layers are set to trainable with the intention of optimization of the already trained layers. Also, extra layers are provided to improve feature extraction. In particular, two Conv2D layers with 64 and 128 filter strength are incorporated, with each of them including a MaxPooling2D layer for downscaling the feature maps. A GlobalAveragePooling2D layer is applied again before the top layer of the model starts in order to minimized the size. The output layer still remains as a Dense layer, which is softmax activated, for classification. The model is then recompiling with same Adam optimizer and loss function and training is continued for 10 more epochs. At this stage, the model is given the ability to tune not only the pre-trained weights but also the extra layers, which gives an opportunity to better include the specific features of the domain and raise the performance.

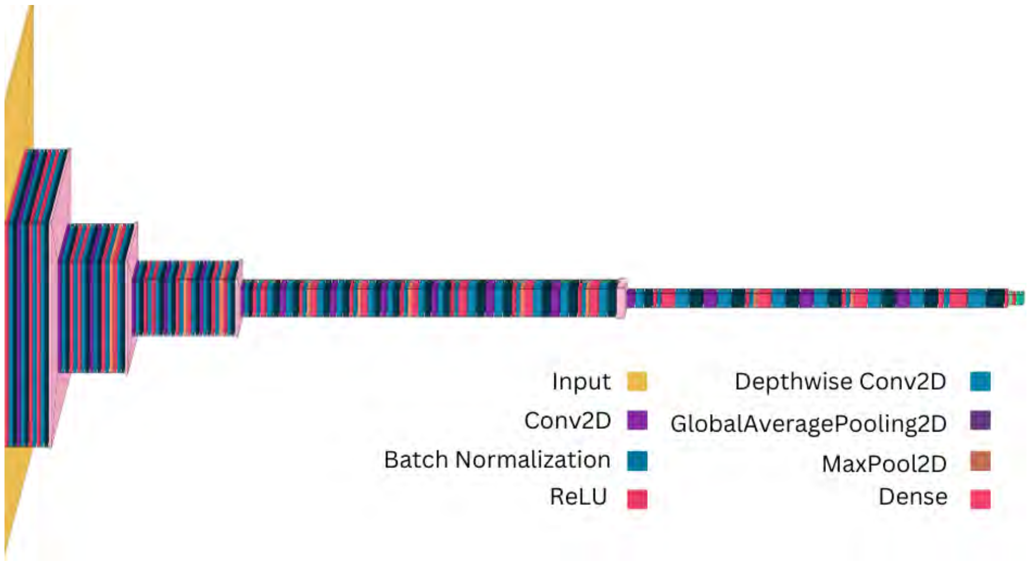

Figure 3.7: Architecture of Custom MobileNetV2

conv2d_4 (Conv2D)	(None, 7, 7, 64)	737,344	out_relu[0][0]
max_pooling2d_2 (MaxPooling2D)	(None, 3, 3, 64)	0	conv2d_4[0][0]
conv2d_5 (Conv2D)	(None, 3, 3, 128)	73,856	max_pooling2d_2[0][0]
global_average_pooling2d… (GlobalAveragePooling2D)	(None, 128)	0	conv2d_5[0][0]
dense_4 (Dense)	(None, 2)	258	global_average_poolin…

Figure 3.8: Architectural Summary of Layers Added

Chapter 4

Result Analysis & Discussion

4.1 Result Analysis

For the purpose of carrying out the model's training, certain parameters were predetermined at the onset of the process. The batch size of the models was set to 16, while 10 was the number of epochs established for each model. The performance of each model in terms of accuracy and loss is given in the Table 4.1.

Table 4.1: Training and Validation Loss and Accuracy of the models

Model	Training		Validation	
	Accuracy	Loss	Accuracy	Loss
VGG16	0.9050	0.2735	0.9161	0.2564
VGG19	0.8979	0.3126	0.9007	0.2943
InceptionV3	0.9562	0.1265	0.9519	0.1299
MobileNetV2	0.9628	0.1048	0.9529	0.1219
DenseNet121	0.9531	0.1295	0.9550	0.1281
Custom MobileNetV2	0.9923	0.0198	0.9857	0.0904

Table 4.1 presents the performance metrics of six models, including VGG16, VGG19, InceptionV3, MobileNetV2, DenseNet121, and a custom MobileNetV2, evaluated on both training and validation sets. VGG16 achieved 90.50% accuracy in training and 91.6% in validation with a moderate loss of 0.2735 and 0.2564, respectively, showing a good balance between training and validation performance. VGG19, while close to VGG16, had lower accuracy (89.79% training and 90.07% validation) and higher losses, indicating it might require further tuning. InceptionV3 performed strongly with over 95% accuracy in both sets and low loss values (0.1265 in training and 0.1299 in validation), showing excellent generalization. MobileNetV2 followed closely with 96.28% training accuracy and 95.29% validation accuracy, coupled with low loss values, highlighting its robust performance. DenseNet121 showed similarly high accuracy, with 95.31% in training and 95.50% in validation, indicating solid generalization and consistency. The custom MobileNetV2 model significantly outperformed all other models, reaching 99.23% training accuracy and 98.57% validation accuracy, with the lowest loss values (0.0198 in training and 0.0904 in validation). This suggests the custom model is highly optimized and generalizes well across the

dataset. The consistency between training and validation metrics across most models indicates little overfitting, except for slight tuning needs in the VGG models. Overall, the custom MobileNetV2 is the best-performing model in this comparison, while InceptionV3, MobileNetV2, and DenseNet121 also show strong results.

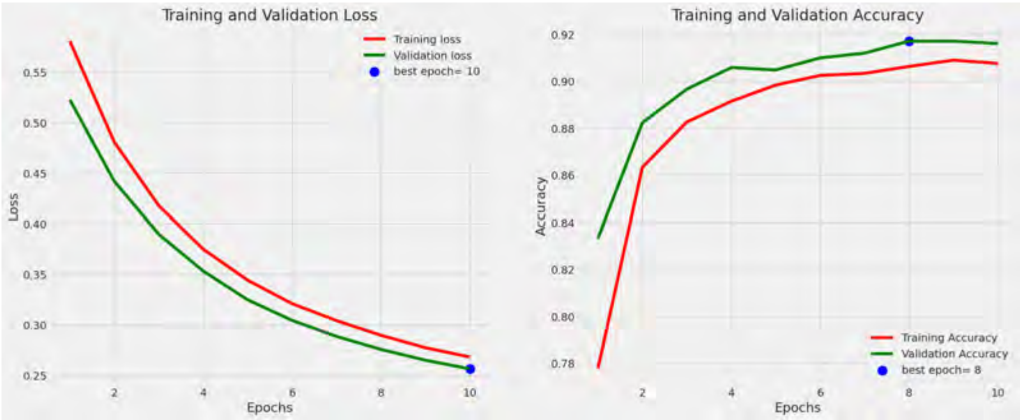

Figure 4.1: Training and Validation Accuracy and Loss Curve for VGG16

The figure shows the training and validation loss (left) and accuracy (right) curves for the VGG16 model over 10 epochs. Both the training and validation loss decrease consistently with increasing epochs, indicating improved model performance and convergence. The validation loss remains slightly lower than the training loss, suggesting good generalization. On the accuracy side, both training and validation accuracy increase steadily, with the validation accuracy slightly outperforming the training accuracy after around 7 epochs, which is a positive sign of minimal overfitting. The best epoch is marked at epoch 8 for accuracy, where the model achieves its peak validation performance.

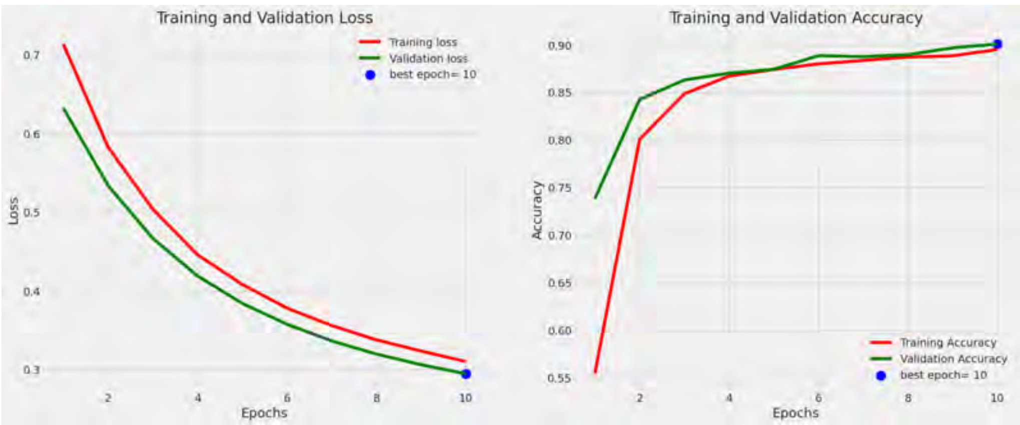

Figure 4.2: Training and Validation Accuracy and Loss Curve for VGG19

The image depicts the loss (left) and accuracy (right) curves for training and validation processes of the VGG19 model across 10 epochs. In both training and validation processes, the loss continues to drop, which shows the enhancement in the model, although the validation loss is observed to be generally lower than the training loss. In the curves showing the accuracy for training and validation, both rush upward in the first few epochs with a resort after around 4 epochs. Training accuracy is lower than validation accuracy, which indicates training generalization. The final results are presented for epoch 10 where both accuracy and loss shown for validation are at their best.

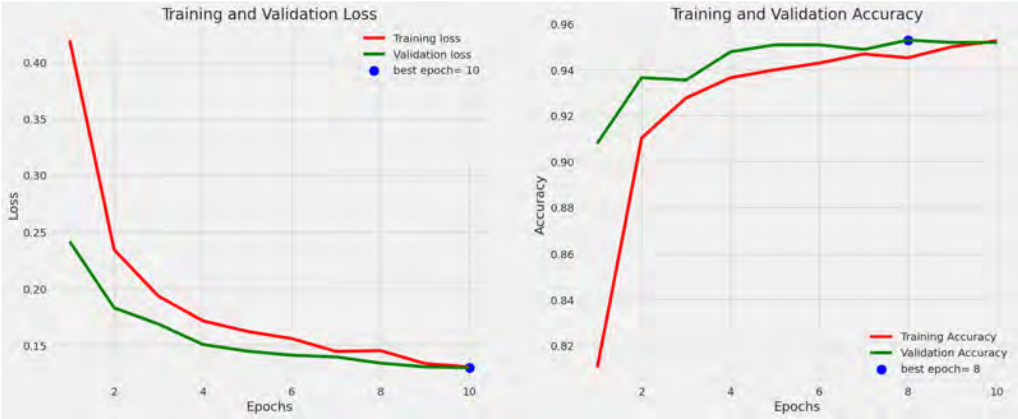

Figure 4.3: Training and Validation Accuracy and Loss Curve for InceptionV3

This image illustrates the loss (left) and accuracy (right) curves for training and validation of the InceptionV3 model over 10 bulbs. Both the training and validation loss exhibit a clear decrease especially at the beginning epochs and then flatten out at the later epochs showing that learning has taken place and convergence has been reached. It can be noted that the validation loss remains on the average lower than the training loss which is a positive indicator. For accuracy, both training accuracy and validation accuracy obs reveal an increase and level off around epoch 5 where validation accuracy remains moderately higher than the training accuracy. The last epoch to demonstrate sensitivity is assigned epoch 8 when the model lets the accuracy impressive performance.

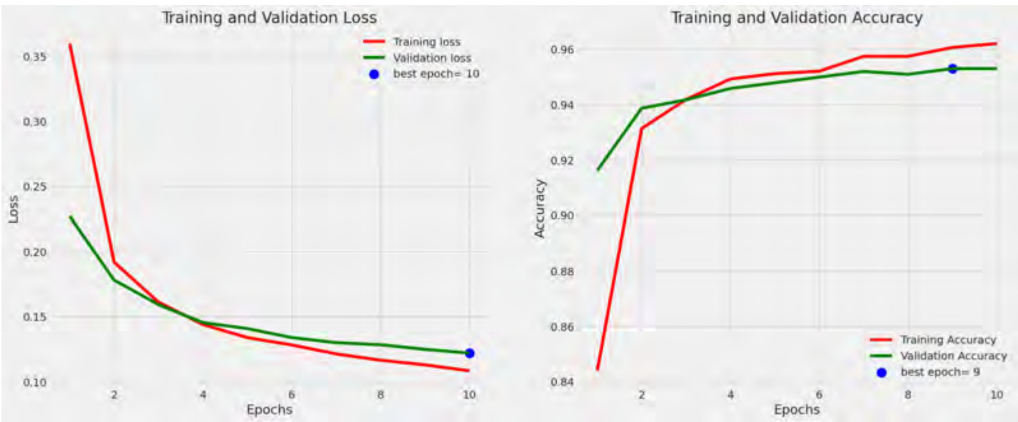

Figure 4.4: Training and Validation Accuracy and Loss Curve for MobileNetV2

The diagram represents the MobileNetV2 model's training and validation loss (to the left) and accuracy (to the right) graphs over 10 epochs. And both the training and validation loss are decreasing throughout the epochs, and the validation loss is lower than the training loss, signifying proper learning and healthy overfitting. The accuracy curves exhibit a sharp upturn in the beginning epochs, while both training and validation accuracies plateau after the fourth epoch, that too at quite a high level. And training and validation accuracies are almost the same indicating a good generalization of the model with not much overfitting. The model showcased its optimal performance at epoch 9 where the highest validation accuracy was noted.

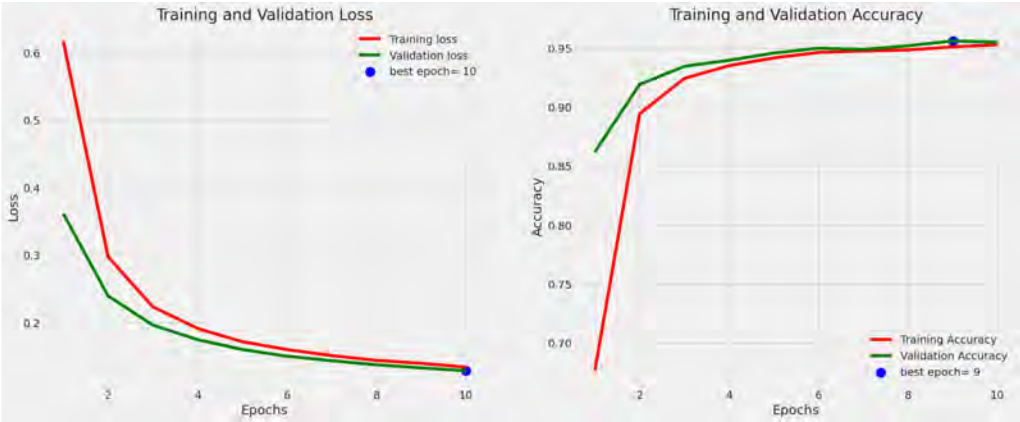

Figure 4.5: Training and Validation Accuracy and Loss Curve for DenseNet121

The graph illustrates the loss (left side) and accuracy (right side) of the DenseNet121 model at the end of 10 epochs training. The graphs indicate that both the training and the validation losses decrease over epochs, further the validation loss is consistently lower than the training loss depicting good generalization. The accuracy curves exhibit rapid growth in the first several epochs; this is followed by training and validation accuracy curves that remain flat after epoch 4, the two accuracy measures achieving greater than 95% values. The high degree of resemblance between the training accuracy curve and validation accuracy curve indicates that the model has not overfit optimally. The highest value of validation accuracy of the model occurs at epoch 9, which signals the highest achievable performance of the model.

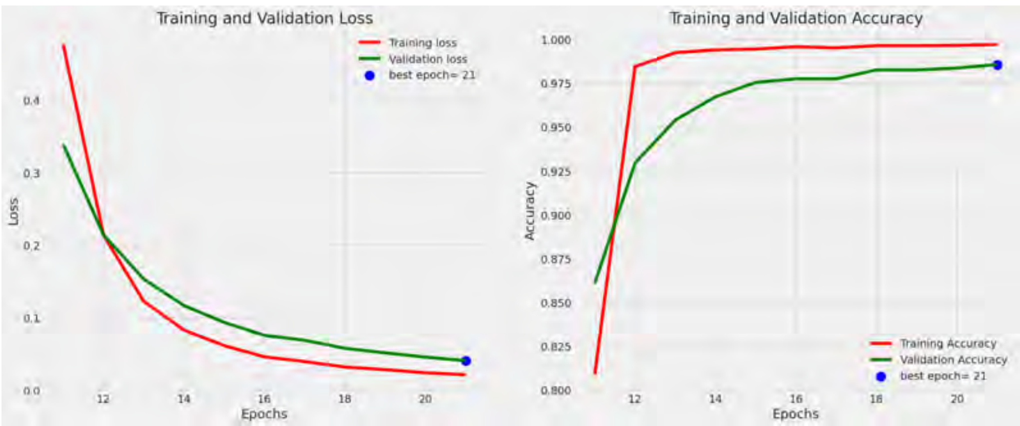

Figure 4.6: Training and Validation Accuracy and Loss Curve for Custom MobileNetV2

The graphic showcases the custom MobileNetV2 model's performance across 21 epochs, (left) displaying loss and (right) accuracy. Training loss decreases consistently, reflecting the model's learning improvement, while validation loss fluctuates initially but stabilizes towards the later epochs, with the 21st epoch being optimal. Both training and validation accuracy curves show strong performance with minimal divergence. Although there's a minor dip in validation accuracy mid-training, it recovers, demonstrating the model's effective generalization and stable training.

Table 4.2: Precision, Recall, F1 Score and Accuracy of the Models

Model	Precision	Recall	F1 Score	Accuracy
VGG16	0.92	0.92	0.92	0.92
VGG19	0.90	0.90	0.90	0.90
InceptionV3	0.95	0.95	0.95	0.95
MobileNetV2	0.96	0.96	0.96	0.96
DenseNet121	0.95	0.94	0.94	0.94
Custom MobileNetV2	0.98	0.98	0.98	0.98

The table assesses and compares six models based on a few performance indicators, including precision recall, F1 score, and accuracy. A plausible performance is demonstrated by VGG16 model with a precision recall F1 score and accuracy of 0.92, demonstrating fairness in all these measures. All the metrics of VGG19 are slightly lower than that of VGG16 model (0.90), proving that VGG19 is arguably weaker than VGG16 in this case. InceptionV3 has also proven to be a reliable and consistent model with a precision recall and F1 score of 0.95. The previous models are surpassed by MobileNetV2 in all respects with a precision recall and F1 score of 0.96, meaning that it was even better for that particular classification problem. DenseNet121 also performs fairly well with a precision of 0.95 and recall of 0.94, however its F1 (0.94) and accuracy (0.94) figures fall slightly short of distance of MobileNetV2. Custom MobileNetV2 emerges the best model with 0.98 of precision, recall, F1 score, and accuracy, which reflects nearly a hundred percent classification ability. Since the metrics are similar across the board, it means all the models are well optimized with the exception of predictive performance where the custom MobileNetV2 stands tall. In summation, the custom MobileNetV2 is the best performer but other models like MobileNetV2 and InceptionV3 also yield good results. VGG19 fails to catch up further demonstrating the capabilities of new generation architectures compared to the older ones.

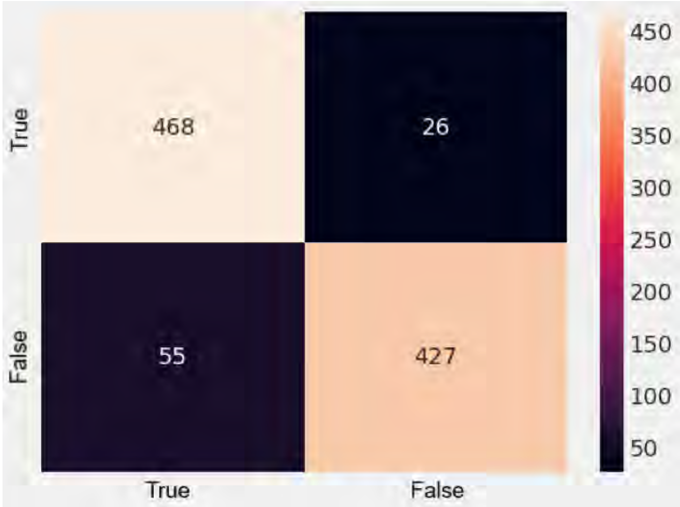

Figure 4.7: Confusion Matrix for VGG16

The Visualization and Generalization 16 deep learning model performance on the tested dataset is also presented using a confusion matrix accurate and false classification counts. True was classified correctly 468 instances and 427 instances as False. There are 26 false predictions, issuing a True label where it is actually a False while 55 cases of omission were recorded where the system issued a False label but it was a True affirmation. This matrix reveals that the false positive and the false negative error rates associated with the VGG16 model are not far off from each other, although the former seems to be a bit lower than the latter. The effectiveness of the model is underscored by the sheer number of images that were correctly classified.

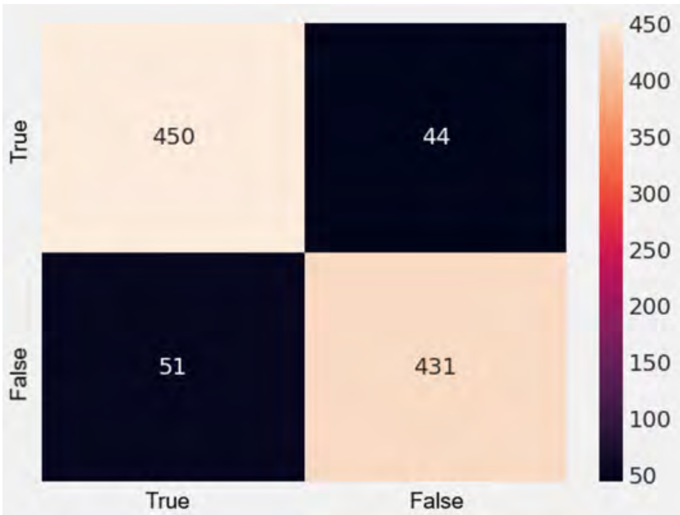

Figure 4.8: Confusion Matrix for VGG19

The confusion matrix that the VGG19 Network has evaluated reveals the model performance of classification. The model has made 450 correct predictions in support of “True” and 431 in support of “False”. There are 44 instances of false positive where the model incorrectly labeled a ‘False’ instance as ‘True’ and 51 instances of false negative where the model labeled a ‘True’ instance as ‘False’. This matrix suggests that VGG19 has a slightly higher cost for false positives compared to VGG16, however the number of false negatives is comparatively the same. The accuracy of this model is high nevertheless it has a more or less high error rate than the VGG16 model due to misclassifying true calls as false.

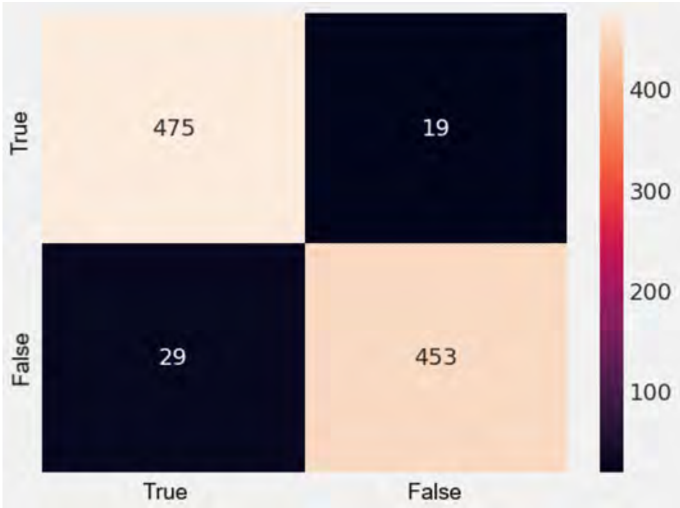

Figure 4.9: Confusion Matrix for InceptionV3

The InceptionV3 model's performance in classification is summarized visually by means of the confusion matrix, where ”True” was correctly identified in 475 instances, while ”False” was identified correctly in 453 instances. There are few false positives (19) where the model was incorrect in saying “True” for something that was a “False” and there are also a few false negatives (29) where a “False” was given to a “True” instance. Inauguration of VGG models was marred by a lot of errors which were not present in Inception V3 proving that it had greater precision and recall as shown by the smaller number of false positives and false negatives. The overall performance of the model is excellent, as it achieves a high level of accuracy per both classes with very few instances of misclassification.

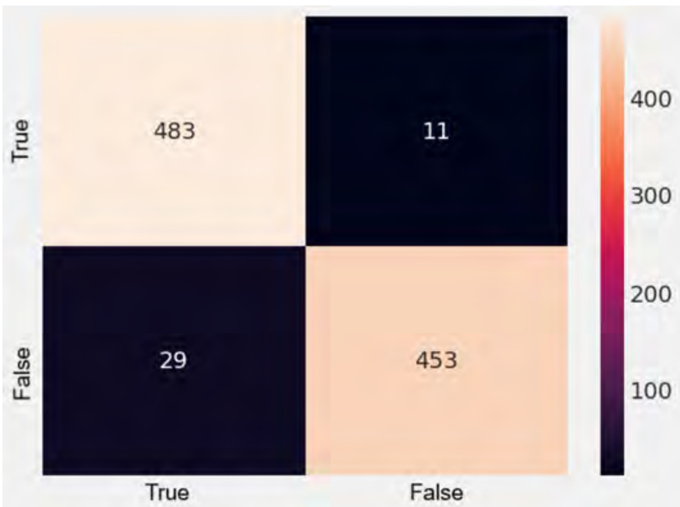

Figure 4.10: Confusion Matrix for MobileNetV2

The MobileNetV2 model's confusion matrix indicates that the model operates quite well in terms of classification, as 483 instances were classified correctly as ”True” while 453 were classified correctly as ”False.” The model produces very few false positives (11), which shows that it is precise in predicting ”True” cases, and there are also ending false negatives, 29, thus good recall as well. In comparison to other models such as VGG and InceptionV3, MobileNetV2 has the smallest number of false-positive instances which proves that predicting ”False” is very accurate for this model. The generally low rates of misclassification observed imply that MobileNetV2 performs well in the datasets it has been trained on, demonstrating both high accuracy and reliability in the predictions made for both classes.

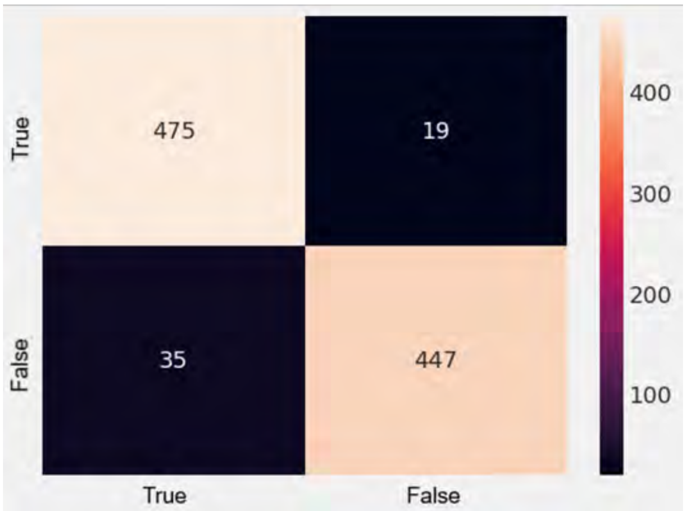

Figure 4.11: Confusion Matrix for DenseNet121

The confusion matrix corresponding to the DenseNet121 model indicates that 475 instances were classified correctly as ‘True’ and there were 447 cases that were marked ‘False’ without an error. There are 19 cases of false positive where instances that should have been ‘False’ were classified by the model as ‘True’ and 35 cases of false negative where the instances which were supposed to be ‘True’ were classified ‘False’ by the model. When looking at other models, it is evident that DenseNet121 has a few more false negatives which suggests to some extent the model tends to have trouble with ‘True’ class predictions faces when compared to less optimized models like MobileNetV2. Regardless of that, the overall performance of the model is good with fairly low rates of misclassification. The small number of false positives shows a high precision whereas the few false negatives would mean that there is some sacrifice in terms of recall for this model.

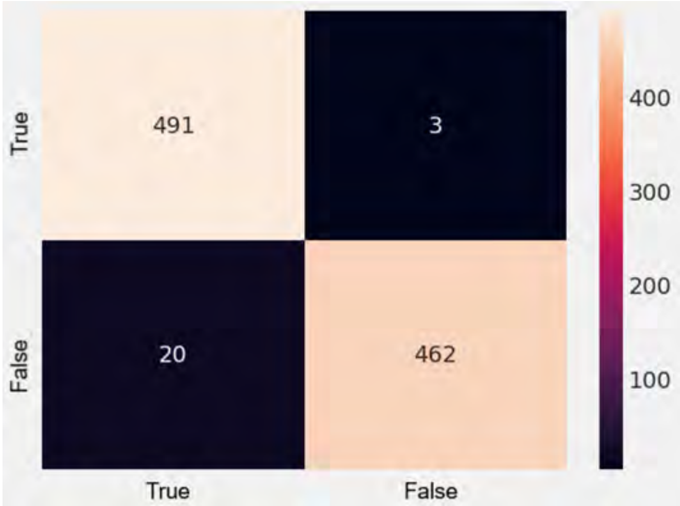

Figure 4.12: Confusion Matrix for Custom MobileNetV2

The confusion matrix for the Custom MobileNetV2 model indicates outstanding classification quality, with 491 true instances classified as ”True” and 462 as ”False.” There are 3 false positives in the model, which means the precision of identifying “True” instances is very good. There are 20 false negatives which means the model has strong recall but there is still a slight room to improve on identifying ”True” instances. In comparison to the other models, Custom MobileNetV2 records the least number of misclassifications most especially false positive cases where the overall accuracy is high. The characteristics of the model which has low false positive cases and in very few cases cannot opportunities due to the determination of the performance reveals that the model generalizes to the cut-offwell. This model is significantly better than others with regards to precision, recall, and overall classification performance.

Table 4.3: Precision, Recall and F1 Score for each class in each model (Rounded to 2 decimal places)

Model	**Class**	**Precision**	**Recall**	**F1 Score**
VGG16	Malnutrition	0.89	0.95	0.92
VGG16	Nutrition	0.94	0.89	0.91
VGG19	Malnutrition	0.90	0.91	0.90
VGG19	Nutrition	0.91	0.89	0.90
InceptionV3	Malnutrition	0.94	0.96	0.95
InceptionV3	Nutrition	0.96	0.94	0.95
MobileNetV2	Malnutrition	0.94	0.98	0.96
MobileNetV2	Nutrition	0.98	0.94	0.96
DenseNet121	Malnutrition	0.93	0.96	0.95
DenseNet121	Nutrition	0.96	0.93	0.94
Custom MobileNetV2	Malnutrition	0.96	0.99	0.98
Custom MobileNetV2	Nutrition	0.99	0.96	0.98

The performance of six different models, specifically VGG16, VGG19, InceptionV3, MobileNetV2, DenseNet121, and Custom MobileNetV2 has been evaluated on two classes, Malnutrition and Nutrition, using different metrics - precision, recall, and F1 score. Custom MobileNetV2 scored the highest with almost perfect classification scoring an F1 score of 0.98 in both classes signifying both high precision and recall. MobileNetV2 also performed well achieving balanced scores of 0.96 in both classes. The InceptionV3 came next with F1 scores of 0.95 in both classes demonstrating dependable classification. DenseNet121 achieves slightly lower scores than Inception but it is also high with 0.95 in Malnutrition and 0.94 in Nutrition. On the other hand, both VGG16 and VGG19 achieved slightly lower scores, though still effective, averagely scoring 0.92 and 0.91 for VGG16 and scoring 0.90 for both classes in VGG19 showing low precision and recall. All in all, it can be noted that later models, notably Custom MobileNet V2, do better than the earlier VGG models in the classification task.

4.2 Model Evaluation

The model's execution was surveyed using Precision, Recall, F1-score. Precision may be a degree that calculates the extent of accurately distinguished positive perceptions out of all the anticipated positive perceptions. Equation: Precision is decided by isolating the number of accurately recognized positive perceptions by the full number of anticipated positive perceptions, as appeared with the taking after equation.

$$Precision = \frac{TruePositives}{TruePositives + FalsePositives} \tag{4.1}$$

Recall, often referred to as sensitivity or true positive rate and is determined by dividing the number of true positive observations by the total number of genuine positive cases.

$$Recall = \frac{TruePositives}{TruePositives + FalseNegatives} \tag{4.2}$$

The F1-score is a statistical measure that represents the harmonic mean of precision and recall. It provides a balanced evaluation of both metrics.

$$F1 \text{ Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{True Positives} + \text{False Positives}} \tag{4.3}$$

4.3 Discussion

The data provided in the table summarizes the evaluation of six distinct models—VGG16, VGG19, InceptionV3, MobileNetV2, DenseNet121, and Custom MobileNetV2—for two categories: Malnutrition and Nutrition. The analyzed indicators are precision, recall, and F1 score, all of which form a large picture of the classification's effectiveness.

In the range of the tested models, Custom MobileNetV2 is perefomring the best with almost ideal precision, recall, and an F1 score of 0.98 in classifying both Malnutrition and Nutrition. This shows that there are very few cases of the model incorrectly classifying or identifying the two variables making it the best model for the study. MobileNetV2 gives good results too, scoring 0.96 F1 in both classes indicating that the foundation architecture of MobileNetV2 is efficient in classifying these health categories. These results imply that the MobileNet model types are very apt for the used dataset because both versions provide an excellent precision-recall trade-off.

The prowess of InceptionV3 is also commendable with 0.95 F1 score for both classes which indicates excellent generalization and reliability. DenseNet121 is a close second recording 0.95 for Malnutrition classification and 0.94 for Nutrition classification. Although performing consistent and dependable classification, it seems to perform lower than MobileNetV2 and InceptionV3 when classifying precision for the Nutrition class. This is depicted by the results of the models as it appears that they have been optimized to perform classification in the presence of a well balanced dataset. On the other hand, older designs such as VGG16 and VGG19 do not perform as well as the newer ones. VGG16 registers 0.92 and 0.91 scores in the Malnutrition and Nutrition categories while VGG19 scores 0.90 in both cases. These lower results signify that even though VGG models are still fairly capable of performing, they are lesser suited for this dataset in comparison to advanced design such as MobileNetV2 and InceptionV3.

Consequent to the findings, it appears that more advanced architectural models and new enhancements such as MobileNetV2 and Custom MobileNetV2 are superior to the older VGG models. This difference in performance is probably due to the improved effectiveness and features a model newer than VGG possesses that make it ideal for classification problems with closer boundaries such as Malnutrition and Nutrition. Among all ml models trained, Custom MobileNetV2 was observed to outperform all other models thanks to its superiority in precision, recall, and F1 score. This makes it the best fit for the addressed classification problem.

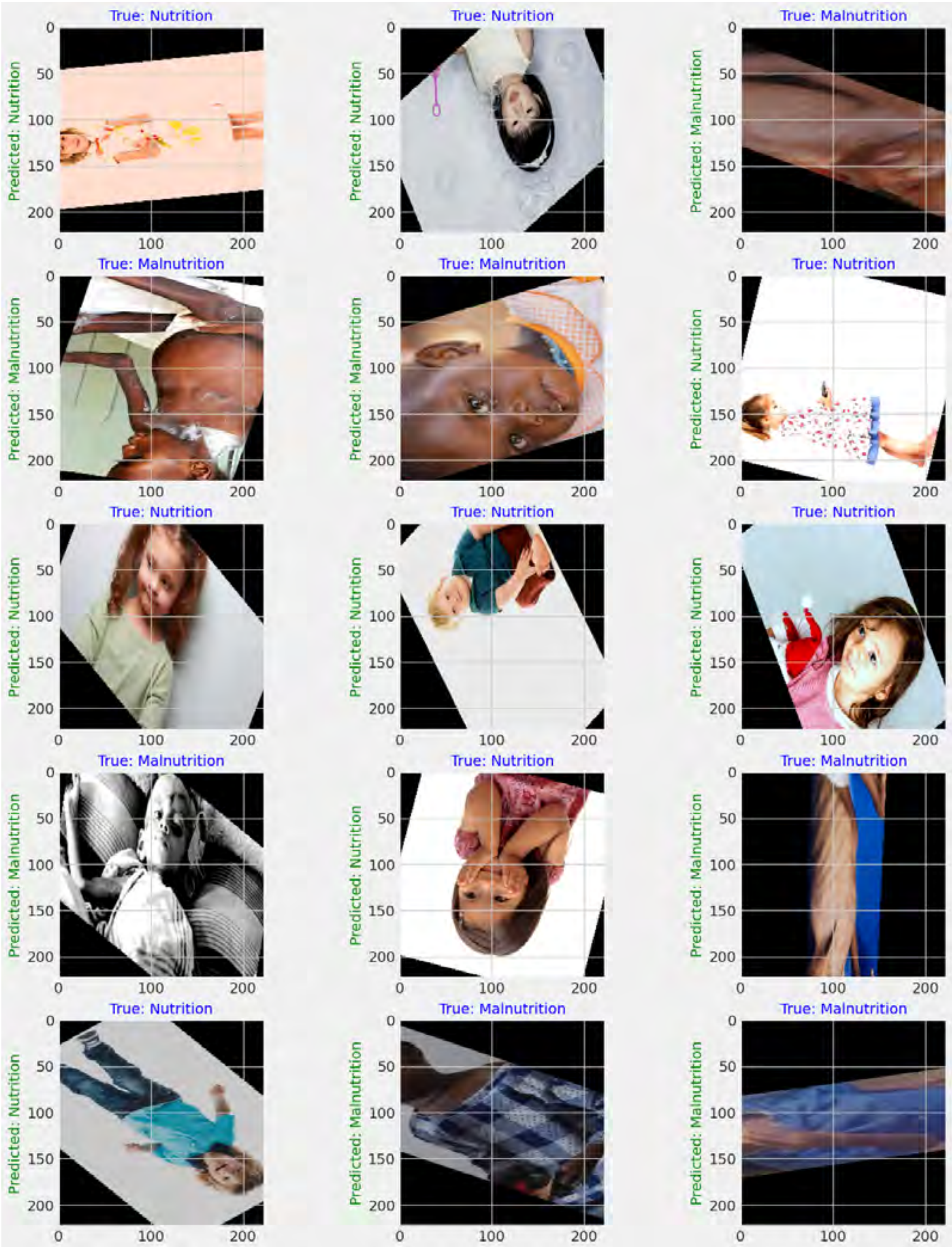

Figure 4.13: Prediction of Custom MobileNetV2

Chapter 5

Conclusion & Future Work

5.1 Conclusion

Children's malnutrition can potentially and automatically be detected looking at features of their facial structures through the use of advanced deep learning architectures. Rigorous testing and assessment of VGG16, VGG19, InceptionV3, MobileNetV2, DenseNet121, and even a custom MobileNetV2 showed that deep learning models can classify malnourished and healthy children with alarming accuracy. Among the models, custom MobileNet V2 gave the best performance with a training accuracy of 99.23% and a validation accuracy of 98.57%, thus proving its effectiveness in malnutrition detection systems. It was also possible to show how adding transfer learning, data enhancement, and careful training of specific models may lead to improved performance of these systems in an automated way. The results turned out to give a very good basis regarding how AI can be applied in healthcare diagnostics whereby it becomes easier to detect such children with high precision in an expeditious manner thereby allowing treatment in good time.

5.2 Future Work

Furthermore to the contributions made in this study on implementation of deep learning in detection of malnutrition, there are other areas to explore in future studies. First, expanding the scope of the dataset in terms of images, especially those depicting different ethnicities and environmental settings would improve the trained model's applicability. In addition, XAI can be added to increase the understandability and acceptability of the model to health care workers in that they will connect with how the decisions are made. However, the dataset adopted for the present research is balanced with no bias further enhancing the accuracy of model forecasting performance across many population groups. Moreover, improving the dataset and implementation of more advanced image alteration strategies could improve results on the smallest classes. Further attempts may also include incorporation of additional health dimensions (for instance, height, weight and body mass index) into the image of mapping malnutrition by face detection to reach the standard used in other health sciences like Nutrition. Health education combined with facial images and medical statistics will be more precise in terms of detection efficiency.

Bibliography

[1] V. Bagilkar and B. Savadatti, "A descriptive study on malnutrition," *Asian Journal of Nursing Education and Research*, vol. 5, Jan. 2015. DOI: 10.5958/2349-2996.2015.00016.6.

[2] N. Shukla, N. Toppo, A. Thakur, and P. Kasar, "A study of malnutrition and associated risk factors among children of age 06-59 months in urban area of jabalpur district (m.p.)," *International Journal Of Community Medicine And Public Health*, vol. 5, p. 296, Dec. 2017. DOI: 10.18203/2394-6040.ijcmph20175801.

[3] K. Dong, C. Zhou, Y. Ruan, and Y. Li, "Mobilenetv2 model for image classification," in *2020 2nd International Conference on Information Technology and Computer Application (ITCA)*, 2020, pp. 476–480. DOI: 10.1109/ITCA52113.2020.00106.

[4] C. Dhanamjayulu, N. U. N, P. K. R. Maddikunta, *et al.*, "Identification of malnutrition and prediction of bmi from facial images using real-time image processing and machine learning," *IET Image Processing*, vol. 16, pp. 647–658, 3 2021. DOI: 10.1049/ipr2.12222.

[5] H. M. Fenta, T. Zewotir, and E. K. Muluneh, "A machine learning classifier approach for identifying the determinants of under-five child undernutrition in ethiopian administrative zones," *BMC Medical Informatics and Decision Making*, vol. 21, no. 1, p. 291, Oct. 2021, ISSN: 1472-6947. DOI: 10.1186/s12911-021-01652-1.

[6] A. R. Lakshminarayanan, P. B, R. V, S. Parthasarathy, A. A. Azeez Khan, and K. Javubar Sathick, "Malnutrition detection using convolutional neural network," in *2021 Seventh International conference on Bio Signals, Images, and Instrumentation (ICBSII)*, 2021, pp. 1–5. DOI: 10.1109/ICBSII51839.2021.9445188.

[7] S. Mascarenhas and M. Agarwal, "A comparison between vgg16, vgg19 and resnet50 architecture frameworks for image classification," in *2021 International Conference on Disruptive Technologies for Multi-Disciplinary Research and Applications (CENTCON)*, vol. 1, 2021, pp. 96–99. DOI: 10.1109/CENTCON52345.2021.9687944.

[8] S. Jain, T. Khanam, A. J. Abedi, and A. A. Khan, "Efficient machine learning for malnutrition prediction among under-five children in india," pp. 1–10, 2022. DOI: 10.1109/DELCON54057.2022.9753080.

[9] K. Jahan, "Rising commodity prices will increase malnutrition," *The Business Standard*, Feb. 2023.

[10] S. K V and S. Thripurala, "Real-time facial emotion detection system using multimodal fusion deep learning architecture," in *2023 International Conference on Electrical, Electronics, Communication and Computers (ELEXCOM)*, 2023, pp. 1–6. DOI: 10.1109/ELEXCOM58812.2023.10370457.

[11] R. Lakshmi and V. Vijayalakshmi, "Face feature analysis for facial paralysis detection caused by acute stroke through densenet121," in *2023 Second International Conference on Advances in Computational Intelligence and Communication (ICACIC)*, 2023, pp. 1–4. DOI: 10.1109/ICACIC59454.2023.10435327.

[12] A. Marisetty, P. R. M, P. Nemani, V. Udutalapally, and D. Das, "Advancing Smart Malnutrition Monitoring: A Multi-Modal Learning Approach for Vital Health Parameter Estimation," *arXiv e-prints*, arXiv:2307.16745, Jul. 2023. DOI: 10.48550/arXiv.2307.16745.

[13] J. Timothy, R. Prasetio, D. Tandi, H. Lucky, and I. A. Iswanto, "Facial emotion recognition with inceptionv3 cnn architecture," in *2023 International Conference on Informatics, Multimedia, Cyber and Informations System (ICIMCIS)*, 2023, pp. 1–5. DOI: 10.1109/ICIMCIS60089.2023.10348994.

[14] "An Innovative Application to Predict Malnutrition and Anemia using ML - Peer-reviewed Journal — ijarcce.com," [Accessed 19-09-2023].