

Classification of Edible and Toxic Mushrooms using Deep Learning

by

**Md. Mahidur Rahman (193014001)
Farhan Al Rashid (202014008)
Tamanna Khanam (193014030)
Farzana Haque Tarin (202014007)**

*Project/Thesis report (CSE499) submitted in partial fulfillment of the
requirements for the degree of*

Bachelor of Science in Computer Science and Engineering

Under the supervision of

Muhammad Golam Kibria, PhD, SMIEEE



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
UNIVERSITY OF LIBERAL ARTS BANGLADESH**

SUMMER 2024

© *Md. Mahidur Rahman, Farhan Al Rashid, Tamanna Khanam and Farzana Haque Tarin*
All rights reserved

DECLARATION

Project Title	Classification of Edible and Toxic Mushrooms using Deep Learning
Authors	Md. Mahidur Rahman, Farhan Al Rashid, Tamanna Khanam and Farzana Haque Tarin
Student IDs	193014001, 202014008, 193014030 and 202014007
Supervisor	Muhammad Golam Kibria, PhD, SMIEEE

We declare that this project/thesis report entitled *Classification of Edible and Toxic Mushrooms using Deep Learning* is the result of our own work except as cited in the references. The project/thesis report has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Md. Mahidur Rahman
193014001

Department of Computer Science and Engineering
University of Liberal Arts Bangladesh

Farhan Al Rashid
202014008

Department of Computer Science and Engineering
University of Liberal Arts Bangladesh

Tamanna Khanam
193014030

Department of Computer Science and Engineering
University of Liberal Arts Bangladesh

Farzana Haque Tarin
202014007

Department of Computer Science and Engineering
University of Liberal Arts Bangladesh

Date: October 20, 2024



Department of Computer Science and Engineering
University of Liberal Arts Bangladesh
Mohammadpur, Dhaka - 1207

CERTIFICATE

This is to certify that the project/thesis report entitled **Classification of Edible and Toxic Mushrooms using Deep Learning**, submitted by **Md. Mahidur Rahman** (Student ID: 193014001), **Farhan Al Rashid** (Student ID: 202014008), **Tamanna Khanam** (Student ID: 193014030) and **Farzana Haque Tarin** (Student ID: 202014007) are undergraduate students of the **Department of Computer Science and Engineering** has been examined. Upon recommendation by the examination committee, we hereby accord our approval of it as the presented work and submitted report fulfill the requirements for its acceptance in partial fulfillment for the degree of *Bachelor of Science in Computer Science and Engineering*.

Muhammad Golam Kibria, PhD, SMIEEE, Professor
Dept. of CSE, University of Liberal Arts Bangladesh

Mohammed Ashikur Rahman, PhD, Assistant Professor
Dept. of CSE, University of Liberal Arts Bangladesh

Rubaiya Hafiz, Senior Lecturer
Dept. of CSE, University of Liberal Arts Bangladesh

Sovon Chakraborty, Lecturer
Dept. of CSE, University of Liberal Arts Bangladesh

Place: Dhaka

Date: October 20, 2024

ACKNOWLEDGEMENTS

We would like to express our deep and sincere gratitude to our research supervisor, *Muhammad Golam Kibria, PhD, SMIEEE*, for giving us the opportunity to conduct research and providing invaluable guidance throughout this work. His dynamism, vision, sincerity and motivation have deeply inspired us. He has taught us the methodology to carry out the work and to present the works as clearly as possible. It was a great privilege and honor to work and study under his guidance.

We are greatly indebted to our honorable teachers of the Department of Computer Science and Engineering at the University of Liberal Arts Bangladesh who taught us during the course of our study. Without any doubt, their teaching and guidance have completely transformed us to the persons that we are today.

We are extremely thankful to our parents for their unconditional love, endless prayers and caring, and immense sacrifices for educating and preparing us for our future. We would like to say thanks to our friends and relatives for their kind support and care.

Finally, we would like to thank all the people who have supported us to complete the project work directly or indirectly.

Md. Mahidur Rahman, Farhan Al Rashid, Tamanna Khanam, Farzana Haque Tarin

University of Liberal Arts Bangladesh

Date: October 20, 2024

Dedicated To
*My beloved parents, the
source of my inspiration*

– Md. Mahidur Rahman

Dedicated to
My loving family

– Farhan Al Rashid

Dedicated To
*My family, for their endless
love and support*

– Tamanna Khanam

Dedicated To
*My mother, my constant
supporter and biggest
inspiration*

– Farzana Haque Tarin

ABSTRACT

Mushrooms have become one of the most profitable businesses in the whole world and with this it is contributing to our GDP. As the demand for mushrooms is increasing, so is the risk for someone to consume toxic or poisonous mushrooms and putting their life at risk. In a study conducted by the World Health Organization it is stated that almost 50,000 people are affected due to mushroom poisoning among which the death rate is in the hundreds. Therefore, we are focusing on the perspective of Bangladesh building awareness and education about the dangers of toxic mushrooms, benefitting both local communities and beyond. It is often difficult to categorize between edible and toxic mushrooms but with the help of machine learning identification has become very easy. To train classification models we have used several machine learning techniques including CNN, VGG 16, ResNet 50, Mobilenet V2, Inception V2, Efficientnet B0 and Hybrid System combining EfficientNetB0 as a feature extractor with a Support Vector Machine (SVM) classifier with accuracies 86%, 87.93%, 91.37%, 93.10%, 92.70%, 86.60% and 99.50%. Among all the models that we have used we can conclude that the most accuracy was obtained by the hybrid model.

Keywords: Mushrooms, Edible, Toxic, Machine Learning, CNN, VGG 16, ResNet 50, Mobilenet V2, Inception V2, Efficientnet B0, Support Vector Machine (SVM), Feature Extraction, Classification.

Contents

1	Introduction	1
1.1	Background Literature	2
1.2	Problem statement	3
1.3	Objectives and Significance of the Study	3
1.3.1	Aims	3
1.3.2	Objective	4
1.3.3	Motivation	4
1.3.4	Significance	4
1.4	Scope and Limitation	5
1.5	Project Planning	5
1.6	Report Layout	6
2	Literature Review	8
2.1	Literature Review	8
2.2	Literature Analysis	10
3	Methodology	11
3.1	Working Process	11
3.2	Proposed Method Framework	12
3.3	Description of the Dataset	13
3.4	Data Preprocessing	13
3.5	Image Resizing	13
3.6	Normalization	14
3.7	Data Augmentation	14
3.8	Train-Test Split	15
3.9	Data visualization	15

3.10	Model Selection	16
3.11	Basic CNN Model	17
3.12	EfficientNetB0 Model	18
3.12.1	Architecture	18
3.12.2	EfficientNetB0 for Feature Extraction	19
3.13	Hybrid Model	20
3.13.1	SVM Model	20
3.13.2	Transfer Learning	20
3.13.3	Feature Extraction Process	21
3.13.4	Model Training	23
3.13.5	Key Features of Support Vector Machine (SVM) for Classifica- tion	23
3.14	Reason Behind Applying Hybrid Model	24
4	Design and Development	25
4.1	Project Management and Financial Activities	25
4.1.1	Financial Report Project Budget	26
4.2	Adopted Tools and techniques	26
4.2.1	Hardware Requirements	26
4.2.2	Software Requirements	27
4.3	Design the Solution	27
4.4	Implementation	29
4.4.1	Import necessary Libraries and Modules	29
4.4.2	Data Processing	31
4.4.3	Model	33
4.5	Evaluate the solution	36
4.5.1	Evaluation Metrics	36
4.5.2	Precision Score	37
4.5.3	Recall Score	37
4.5.4	F1 score	37
4.5.5	Accuracy Score	38
4.5.6	Confusion Matrix	39

5	Result analysis	40
5.1	Model Performance	40
5.2	Class-wise Evaluation	41
5.3	Confusion Matrix	42
5.4	Comparison of Model Performance	44
6	Conclusions	46
6.1	Societal, health, safety, legal and cultural aspects	46
6.2	Impact on Environment and Sustainability	47
6.3	Ethical and professional principles	47
6.4	Brief Summary	48
6.5	Future works	49
	References	50

List of Figures

1.1	Timeline of the project	6
3.1	Overview of the Proposed System	12
3.2	Dataset Distribution (Pie Chart)	15
3.3	Sample from the Dataset	16
3.4	CNN architecture	17
3.5	EfficientnetB0 Model Architecture	19
3.6	Hybrid Extraction Process	21
3.7	Hybrid Model Structure Diagram	22
4.1	Confusion Matrix	39
5.1	Hybrid Plot training and validation loss values and training and validation accuracy values	40
5.2	Hybrid Confusion Matrix	43

List of Tables

2.1	Literature Analysis	10
4.1	Budget Summary	26
4.2	Total cost of Hardware Costs	26
5.1	Hybrid Classification Report (Train)	41
5.2	Hybrid Classification Report (Test)	42
5.3	Comparative analysis of research	44
5.4	Comparative analysis of our own implement model	44

Chapter 1

Introduction

A mushroom is the fleshy fruit body of various fungal groups that cling to ground-surface-growing Basidiomycetes. Scientists think that there are likely many more undiscovered mushroom species than the approximately 14,000 known species. Twenty percent can make someone sick, four percent may test positive, and the remaining one percent can cause someone to die out of the twenty five types of mushrooms that have been found. There are several justifications for eating edible mushrooms even though some of them may not taste very nice. The ability of edible mushrooms to eradicate diseases, cancer cells, and strengthen the human immune system makes them extremely interesting. Like some woody rack mushrooms, inedible mushrooms are too severe to even consider eating. It can also be expelled from the human digestive tract and is indigestible. In addition, many people die from eating deadly mushrooms since they are not aware of their existence.

Eating mushrooms without properly identifying them and confirming their species edibility is not advised. Less than 1% of all mushrooms are known to be toxic, making them dangerous and occasionally lethal species. The ability to distinguish edible mushrooms from inedible and toxic ones is crucial. This work uses computer vision and deep learning to classify mushrooms into edible, inedible, and dangerous categories. Deep learning is an artificial intelligence subfield of machine learning that mimics the way the human brain processes information and creates patterns.

[Zahan et al. \(2021\)](#)

1.1 Background Literature

Mushroom is a fungus that contains dual qualities. Mushrooms are well known as a fancy food item used in several dishes from pizza to pasta, enhancing and enriching their flavors. On the other hand, mushrooms are also recognized for their medical benefits around various parts of the globe. Among the identified edible mushroom species around 700 species are being considered to have medicinal properties [Jo et al. \(2014\)](#). It contains nutrients, antioxidants, anti-inflammatory properties, anti-cancer properties, and some unique compounds that help to boost the immune system, lower cholesterol levels, regulate blood sugars, and so on. Therefore, several diseases that human beings often face can be treated with the particles extracted from edible mushrooms [El-Ramady et al. \(2022\)](#). They are one of the healthiest foods that is grown without any effort.

Although mushrooms have a lot of positive sides, several mushrooms are poisonous and contain deadly toxins which when digested can cause abdominal pain to even death. Mistaking a poisonous mushroom for an edible one can have dire consequences. While there are also few edible mushrooms which contains poisonous compound and the severity of intoxication depends on how much of the mushroom is consumed. On research, it is estimated that among 0.14 million mushrooms worldwide there are 7000 edible mushrooms but the remaining 140000 are not easily identifiable as edible or toxic [Devika & Karegowda \(2021\)](#). Among toxic mushrooms, there are *Amanita phalloides* (Death Cap), *Amanita muscaria* (Fly Agaric), and *Gyromitra esculenta* (False Morel).

As the demand for mushrooms has increased worldwide, a recent statistical analysis is made where it is said that the global production of mushrooms grew from 6.90 to 10.24 million metric tons in the span of 9 years from 2008 to 2019 and based on the latest statistics it is estimated that the global market size of mushroom is valued at USD 50.3 billion in 2021 and is expected to expand to 136 billion within 2032 according to Market.us. Agriculture sector of Bangladesh has been one of the main economic driving forces as they contribute approximately 20% of the national GDP [Zahan et al. \(2021\)](#). In recent years Bangladesh has also taken an interest in mushroom production for national use and exporting them as it is one of the suitable countries for cultivating mushrooms due to its favorable climates with low production cost and the availability to grow substrates [Ferdousi et al. \(2020\)](#).

Bangladesh produces over 40,000 tons of mushrooms every year and its estimated market value is almost around Tk800 crore among the mushrooms Oysters and Milky mushrooms are being produced in large amounts on the country's favorable environment and temperature.

As the demand for mushrooms has increased, people have less knowledge about various kinds of mushrooms and are endangering themselves by consuming poisonous mushrooms. In a study done by the National Library of Medicine in China, their result confirms that around 10,036 outbreaks were recorded among which 38,676 illnesses, 21,967 hospitalizations, and 788 deaths were confirmed between 2010-2020 [Li et al. \(2021\)](#).

1.2 Problem statement

Since Bangladesh's agriculture sector is also focusing on Mushrooms, it has become a trendy food among the people but as there are huge variations of mushrooms around the world with similar structures, it is often impossible to differentiate between them due to the lack of knowledge. So, it is easily possible for people to consume poisonous mushrooms resulting in body discomfort to even death. Our aim is to prevent deaths and save people from consuming poisonous mushrooms like the people did in China. Therefore, identifying the classes is very important for all. The current categorization systems in Bangladesh to identify edible, or toxic mushrooms have a high reliance on expert knowledge and manual identification, which increases the danger of misidentification and poisoning from mushrooms and is very time-consuming. Therefore, we need a system that will help us to identify the mushrooms in their classes on demand.

1.3 Objectives and Significance of the Study

1.3.1 Aims

The main aim of our research is to apply a novel hybrid model that will change agriculture management techniques in Bangladesh. The objective of this research is to create a deep learning model that is accurate and efficient for classifying edible and toxic mushrooms in Bangladesh. The project aims to offer a dependable tool

that can help with mushroom identification, promote sustainable foraging practices, and improve public health and safety by utilising the advantages of EfficientNetB0 for feature extraction and Support Vector Machine (SVM) for classification.

1.3.2 Objective

- To develop a deep learning-based classification system capable of accurately distinguishing between Edible and Toxic mushrooms.
- This system will utilize deep learning techniques to analyze mushroom images and classify them based on their toxicity.
- To improve the accuracy in the systems that is used to separate the toxic and edible ones in their classes

1.3.3 Motivation

- Mushrooms are becoming one of the profitable businesses in Bangladesh and worldwide and it can contribute to grow our national GDP.
- Focusing on Bangladesh specifically means addressing a pertinent issue within the region, potentially aiding in public health initiatives.
- Our work could contribute to building awareness and education about the dangers of toxic mushrooms, benefitting both local communities and beyond

1.3.4 Significance

Although mushrooms are an important natural resource with nutritional and therapeutic properties, misidentifying toxic species can negatively affect one's health, even death. In Bangladesh, where mushroom cultivation is limited, trustworthy techniques to differentiate between edible or dangerous types are desperately needed. Conventional identification techniques depend on specialized knowledge, which the general public may not always have access to. To close this gap, this project uses deep learning to produce a mushroom categorization tool that is accurate, scalable, and efficient.

The project is significant for several reasons:

- Public Health and Safety
- Sustainability in Foraging Practices
- Technological Advancement
- Potential for Broader Applications
- Educational and Research Value

1.4 Scope and Limitation

- **Scope:** This project's main objective is to create a deep learning model that can distinguish between harmful and edible mushrooms that can be found in Bangladesh. The research seeks to develop an accurate and efficient mushroom identification tool by merging Support Vector Machine (SVM) for classification with EfficientNetB0 for feature extraction. A dataset of Bangladeshi-specific mushroom species will be used to train the model, and its performance will be assessed in terms of accuracy, inference speed, and generalisation ability. This technology might be the basis for a real-time application that helps farmers, foragers, and the general public safely identify mushrooms.
- **Limitation:** The dataset for the study is limited to species found in Bangladesh, which could limit its usefulness to other parts of the world. Environmental elements that could affect the accuracy of the model include lighting and the visibility of some mushrooms. Furthermore, even though the model is intended to be extremely accurate, it cannot take the place of knowledgeable mycologists and might have trouble with species whose appearance varies much due to environmental conditions.

1.5 Project Planning

The project planning was carefully designed and analyzed at beginning to ensure that everything could be done within the estimated time. The very first step was to narrow down the field that we wanted to explore and it was done in November.

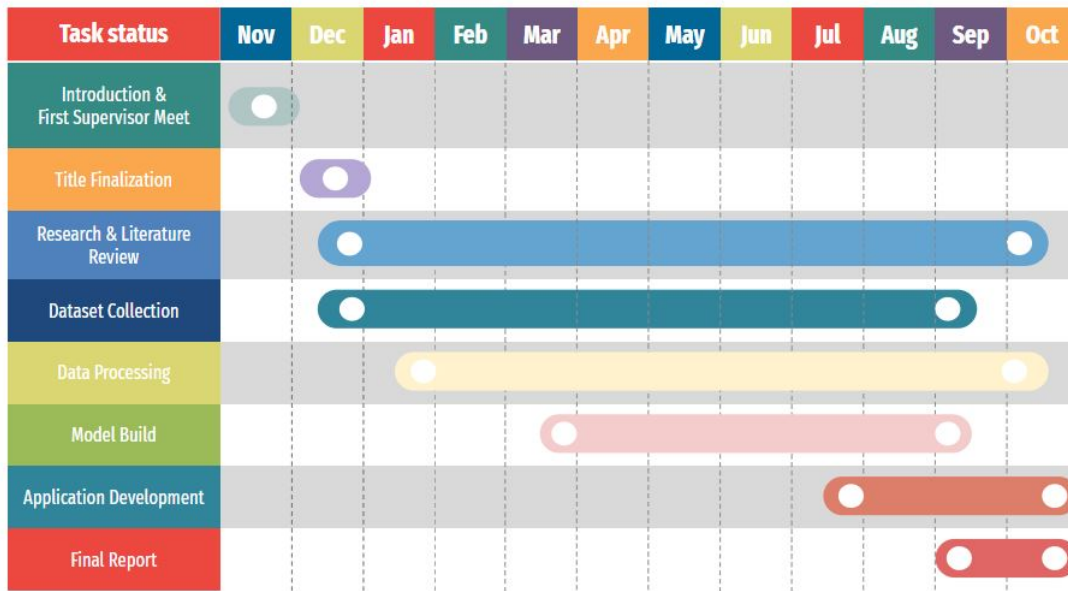


Figure 1.1: Timeline of the project

Following that we carried out our research on the particular field and finalized our project idea in the research and proposal section with extensive research and proposal development that progressed till September. We invested equal amounts of time for collecting and preprocessing data sets which are essential for our project. For the dataset we contacted Mushrooms Development Institute in Savar, Local shops that produce mushrooms and exports them and few online open sources of mushrooms were used. The most important part of the project was implementing the appropriate model which took time from April to august. We selected a few models and implemented them by training and later used a hybrid system to have a comparison analysis between them. Finally, we dedicated our time to comprehensive report writing.

1.6 Report Layout

- **Chapter 1 - Introduction**

An introduction to the world of mushrooms along with their benefits is discussed here. This chapter also focuses on the report's problems, objectives,

and significance. The necessity of the research for public health, technological advancement, and education for the people of Bangladesh.

- **Chapter 2: Literature Review**

This chapter presents a comparative analysis of the literature corresponding to the study. It includes the study of different algorithms that have been developed or can be developed to achieve the objectives of other researchers, giving us a general idea of how we should approach them.

- **Chapter 3: Methodology**

This chapter is about how the data was collected from various sources and how they were processed by pre-processing, resizing, normalization, and so on so that the data are ready for training and testing. A brief summary of the models that are going to be used here is also given. Each step required to develop the model is described and a comparison between several models to analyze the performance is also discussed.

- **Chapter 4: Design and Development**

This chapter focuses on the hardware and software requirements along with the financial budget that is estimated for the development process. Further, the estimated timeline for the entire project is also discussed. Lastly, the implementation of the model is discussed along with the scores.

- **Chapter 5: Result Analysis**

In this chapter, the performance of the hybrid mode with the help of cross-validation and performance metrics and a comparison of the model with other models such as CNN, VGG 16, SVM, and so on are discussed.

- **Chapter 6: Conclusion**

The final chapter summarizes the research findings and discusses the impact of the project on the aspect of social, health, and culture. The project's contribution toward a sustainable environment is also added. Lastly, a summary of the entire project and a few recommendations for further research, focusing on the areas where the system can be improved further for additional challenges is stated.

Chapter 2

Literature Review

2.1 Literature Review

Machine Learning is the field of study that enables computers to learn from data. Deep learning is a part of machine learning where AI is used for supervised, unsupervised as well as reinforcement machine learning. A collection of data is used to train the machine so that it can later sort and identify objects according to their classes easily which is often time consuming for a human to do so. Deep learning uses various types of algorithms such as CNN, ANN, GANs, GNN, KNN and so on. Depending on the field that is being studied, the appropriate algorithms are used. In order to conduct research, we need to analyze various different types of algorithms and then only we can get the output with respect to accuracy and precision [Aravindhaa et al. \(2024\)](#).

Convolutional Neural Networks is a part of machine learning that is used to distinguish between data with the use of image recognition and it is widely used in agriculture field for many purposes [Devika & Karegowda \(2021\)](#), [Shehab et al. \(2023\)](#). CNN resulted significant accuracy and became a standard method in image classification tasks for machine learning after 2012 ILSVRC [Wang et al. \(2022\)](#). Instead of using manually constructed image attributes as classifiers in traditional machine learning approaches, CNNs may automatically learn the distinguishing features of images/objects from pixel arrangements in images [Demirel & Demirel \(2023\)](#). At the current time there are two main types of target recognition algorithm. One is a two-stage algorithm such as R-CNN and the other one is a regression based first order algorithm such as YOLO. YOLOv5 consists of four different models that includes Yolo 5s, Yolo V5m, Yolo V5l and yolo 5x. They are pre-trained with an 80

class MSCOCO dataset [Cengil & Çınar \(2021\)](#).

The authors in [Demirel & Demirel \(2023\)](#) have used about 5500 pictures of mushrooms to test train and validate their model. They have applied CNN-Model as well with the help of "Mobilenetv2.GAP_flatten_fc" which was improvised from "Mobilenet-V2" as it is small and efficient and works with very low memory and processing capacity and it performed with an accuracy of 97.89%. It can be seen being used in leaf disease detection, detection of fruits and so on. In [Moysiadis et al. \(2023\)](#) they have claimed that YOLOv5 is one of the fastest object detection algorithms and their result shows that it can detect and classify mushrooms with an F1-score of up to 76.5% and an accuracy of 70%. Wang and others in [Wang et al. \(2022\)](#) also conducted their work using YOLOv5 where they have mentioned that it fails sometimes to detect the overlapping or the small targets when recognizing mushrooms in experiments. So they had to improve the model so that it would be suitable for the targets which resulted in better performance with up to 99.24% accuracy.

Decision Tree is another supervised machine learning algorithm that is used for classification and regression. It consists of a root node, branches, internal nodes, and leaf nodes, it is arranged hierarchically where each leaf node represents a decision among multiple choices and the branch node represents a decision. It learns to approximate a sine curve by using if-then-else rule [Tutuncu et al. \(2022\)](#). Analyzing the research conducted in [Shehab et al. \(2023\)](#), they have used 23 species of mushroom and after data cleaning and preprocessing and applying various algorithms, it is concluded that decision tree outperforms the other algorithms with an accuracy of 98% and precision of 97.14%.

On the other hand, Bottom and others in [Ottom et al. \(2019\)](#) also conducted similar experiments with different datasets. They compared Neural Network, Decision Tree and KNN. From the collected dataset they extracted the Eigen histogram and parametric features before applying the algorithms to get better results. Another experiment done by TUTUNCU and others in [Tutuncu et al. \(2022\)](#), where more than 8000 images were collected and 22 features were extracted from them to apply decision tree, Naïve Bayes, SVM and AB where Decision tree had an accuracy of 98.82% and precision of 100%.

2.2 Literature Analysis

Reference	InceptionV3	SVM	NaiveBayes	CNN	ResNet 50	MobileNetV2	VGG 16	YOLO v5	DecisionTree
Zahan et al. (2021)				✓	✓		✓		
Aravindhaa et al. (2024)		✓							
Wang et al. (2022)		✓							
Demirel & Demirel (2023)				✓		✓			
Cengil & Çınar (2021)	✓								
Tutuncu et al. (2022)		✓	✓						
Ottom et al. (2019)		✓			✓				
Shehab et al. (2023)				✓				✓	
Ketwongsa et al. (2022)							✓		✓
Rianasari et al. (2022)			✓						

Table 2.1: Literature Analysis

Chapter 3

Methodology

3.1 Working Process

There are several steps involved in finishing the complete project. They are as follows:

1. Data Collection

2. Data Preprocessing:

- Image Resizing: Resize images to a standard size suitable for input into the Efficient Net B0 model.
- Normalization: Normalize pixel values to a common scale.
- Augmentation: Rotating, flipping to increase dataset variability.

3. Data Splitting:

- Train: Train data is used to training the model.
- Validation: Validation data is used to validate the model.
- Test: Test data is used to test the model.

4. Model Architecture:

- EfficientNet B0 as Feature Extractor.
- Support Vector Machine (SVM) for classification

3.2 Proposed Method Framework

This project’s methodology uses a hybrid model of EfficientNetB0 and SVM to classify edible and toxic mushrooms. It entails several important steps. To maintain consistency and improved model performance, mushroom images are first gathered and go through a data preprocessing step where they are normalized and resized to standard dimensions. The data becomes the mushroom dataset used for training and validation after preprocessing.

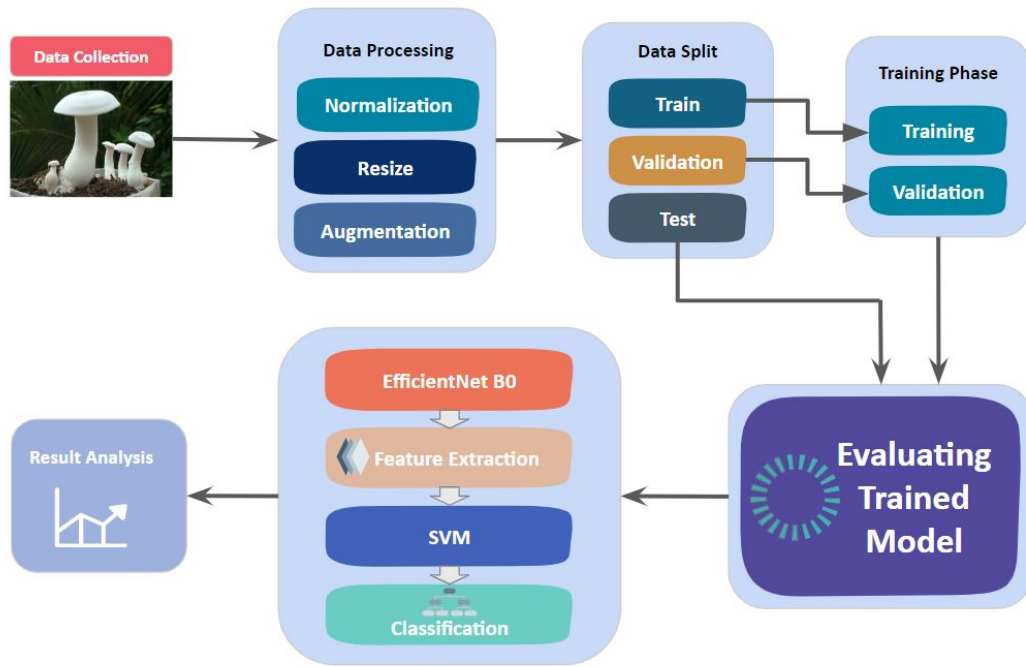


Figure 3.1: Overview of the Proposed System

After that, feature extraction—a critical stage in which pertinent features from the mushroom images are extracted—is carried out using EfficientNetB0. The Support Vector Machine (SVM), which performs the classification task of differentiating between edible and toxic mushrooms, receives these extracted features. To assess the model’s performance and make any required modifications, the classification results are examined. This all-inclusive strategy strikes a balance between efficiency and model interpretability to guarantee correct mushroom classification.

3.3 Description of the Dataset

We have collected different types of mushroom image data from the Mushroom Development Institute (Savar, Dhaka). However, the data we collected was not sufficient enough so we have decided to collect data from open-source (iNaturalist website) as well. There were 6 classes of Mushrooms Agaricus Edible, Agaricus Bisporus Edible, Calocybe Indica Edible, Oyster Edible, Amanita Toxic, and Deathcap Toxic.

3.4 Data Preprocessing

The data preprocessing process for this project involves preparing the mushroom images to ensure consistency and optimal performance of the model. The images are first scaled to a standard size so they can satisfy the EfficientNetB0 model's input size specifications. Normalization, which ensures that all features contribute equally, is done to scale the pixel values after resizing. These values are usually between 0 and 1, which helps stabilize and accelerate the training process. During training and validation, this step improves overall accuracy by reducing computing complexity and accelerating the model's convergence.

Through data preprocessing, we ensure the dependability, variety, and quality of our data. To prepare the photos for later model training and assessment, the preprocessing procedures listed below were carried out.

3.5 Image Resizing

Resizing images is an important aspect of the preprocessing stage in deep learning, especially for models that need uniform input dimensions, like EfficientNetB0. To prevent distortion, the method entails resizing images to a predetermined size while preserving their aspect ratio. Usually, interpolation techniques like nearest neighbor or bilinear interpolation are used to accomplish this, helping to scale the pixel values appropriately. Images are downsized to a common size (e.g., 224x224 pixels) in the context of classifying mushrooms. This ensures consistent input to the model and speeds up training and inference while enabling effective batch

processing. Resizing is crucial for deploying models on low-resource devices since it can also help lower computational load and memory utilization.

3.6 Normalization

Normalization is an essential preprocessing step in deep learning that involves adjusting the pixel values of images to a common scale, typically between 0 and 1 or -1 and 1. By ensuring that every input feature contributes equally to the learning process, this procedure helps enhance the convergence of the model during training. In the context of image data, normalization is frequently carried either by performing mean subtraction and standard deviation scaling to center the data around zero with a unit variance, or by dividing each pixel value by 255 (the maximum value for an 8-bit image). For instance, EfficientNetB0's normalization frequently relies on the ImageNet dataset's mean and standard deviation, which enables the model to make efficient use of pretrained weights. The model can perform better overall during training and inference, achieve faster convergence, and have greater stability when the input images are normalized.

3.7 Data Augmentation

Data augmentation is a powerful technique used in deep learning to artificially increase the size and diversity of a training dataset by applying various transformations to the original images. In order to improve the model's ability to generalize, a greater range of situations and viewpoints are exposed to it through the process of applying random adjustments to the photos, including rotation, scaling, flipping, cropping, brightness adjustment, and color changes. When it comes to the categorisation of mushrooms, data augmentation can improve the model's resilience by mimicking actual differences in mushroom appearance brought about by altered lighting, perspectives, or backgrounds. When the dataset is expanded, the model is less likely to overfit, which improves performance and accuracy when categorizing photos of unseen mushrooms during testing and implementation. The ImageDataGenerator class from the Keras library was used for augmentation during model training.

3.8 Train-Test Split

A dataset is split into two subsets as part of the train-test split process: one for training the model and the other for evaluating its performance. The subsets had an 80:20 split ratio, meaning that 20% datasets were used for testing and the remaining 80% for training. The test set is kept apart to assess how successfully the model generalises to previously unknown data, while the training set is used to teach the model patterns and characteristics. This procedure makes sure that the model is tested on data that it hasn't seen during training, preventing bias or overconfidence in its performance.

3.9 Data visualization

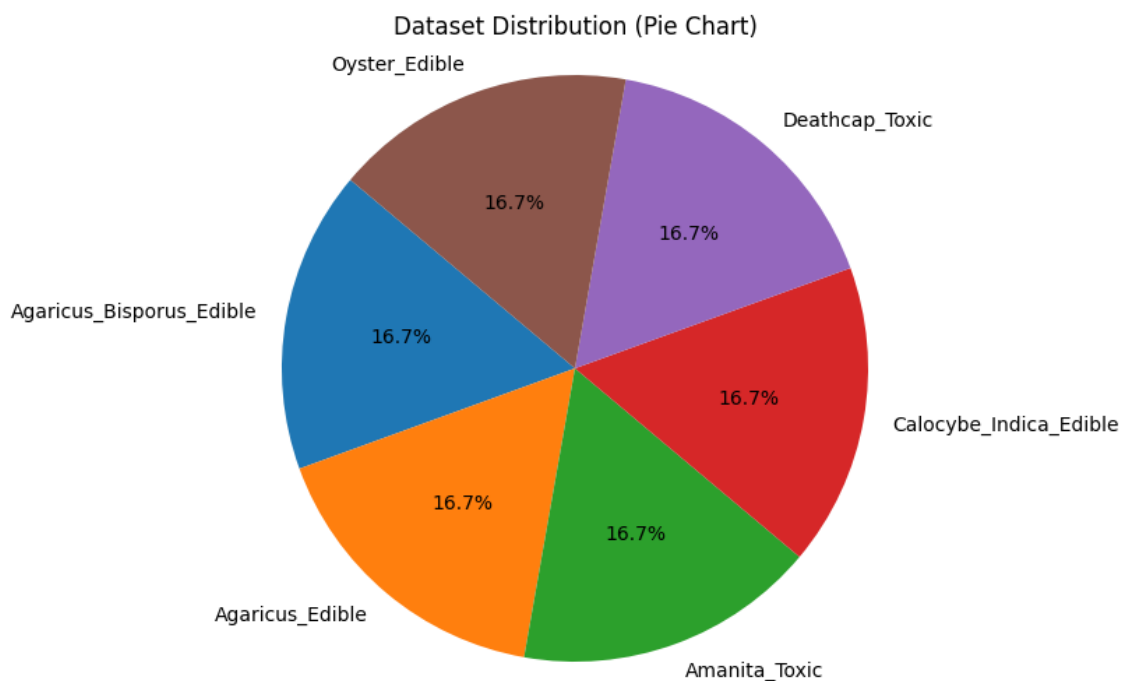


Figure 3.2: Dataset Distribution (Pie Chart)

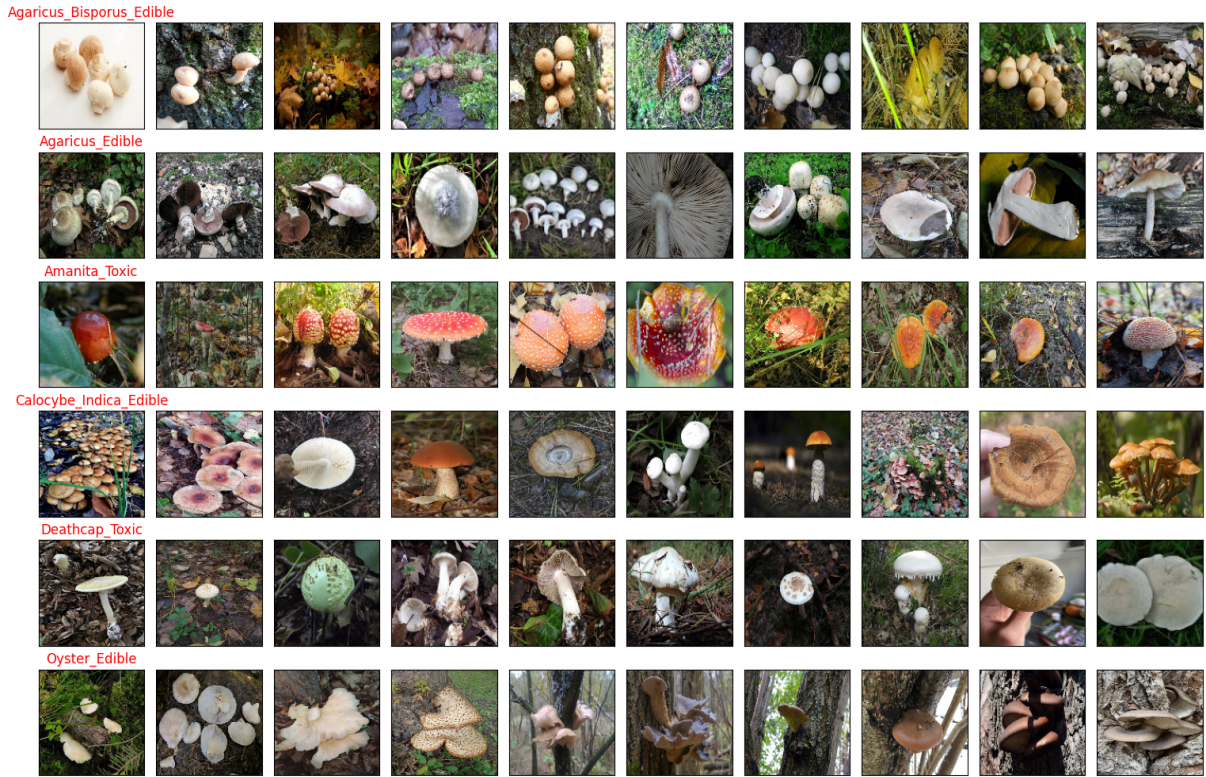


Figure 3.3: Sample from the Dataset

3.10 Model Selection

Following pre-processing of the data, a thorough literature review was conducted once more. To determine the most effective method for developing a hybrid model, a comprehensive investigation of different model designs was carried out. We began with a basic Convolutional Neural Network (CNN) and then investigated more complex models for feature extraction, including VGG16, ResNet50, EfficientNetB0, and other CNN models. Next, we made the decision to select the classification models that would work best for our objective of classifying mushrooms. The models we explored and compared with are as follows:

- CNN
- VGG16
- ResNet50

- Mobilenet V2
- Inception v3
- EfficientnetB0
- SVM

3.11 Basic CNN Model

A basic Convolutional Neural Network (CNN) model is a type of deep learning architecture designed primarily for image classification and recognition tasks. Convolutional layers are the first of several layers that apply filters to the input image in order to identify features such as edges, textures, and patterns. These filters create feature maps that capture various facets of the image by swiping over it. Following convolution, the feature maps are down-sampled using pooling layers (usually max pooling) to lower their dimensionality while keeping crucial information. This lessens overfitting and increases the model's efficiency. The last layers are fully connected layers that analyze the retrieved information and output the prediction, like the class label, in order to function as a classifier. CNNs are very useful for tasks like object detection, picture classification, and visual recognition because they automatically extract hierarchical features from raw image data [Ketwongsa et al. \(2022\)](#).

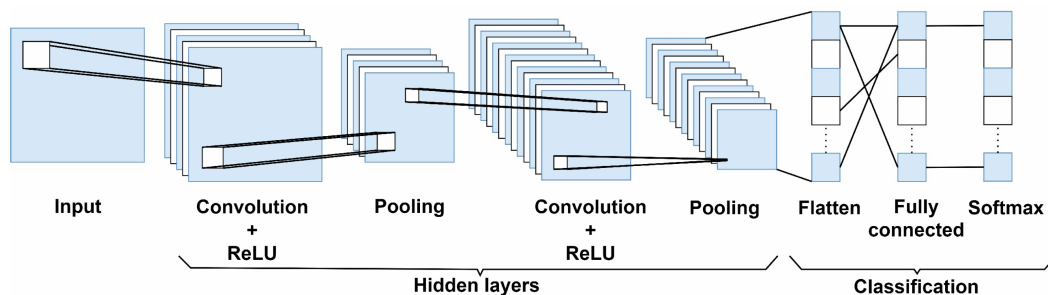


Figure 3.4: CNN architecture

3.12 EfficientNetB0 Model

EfficientNetB0 is a state-of-the-art deep learning architecture designed for image classification tasks, known for its balance between high accuracy and computational efficiency. It is a member of the EfficientNet family, which uses a method known as compound scaling to optimize depth, width, and resolution in order to scale up neural networks in a more methodical manner. Because it is the simplest and most effective model in the series, EfficientNetB0 is perfect for resource-constrained applications. Similar to conventional CNNs, it uses convolutional layers to extract information from pictures; but, because of its more optimized architecture, it offers better performance with fewer parameters and reduced computing costs. This model is lightweight enough for practical deployment in real-world applications, such as mobile and edge devices, and achieves good accuracy on benchmarks like ImageNet.

3.12.1 Architecture

Through a compound coefficient ϕ , EfficientNet's ground-breaking compound scaling technique optimally balances network width (w), depth (d), and resolution (r). By ensuring that the model's dimensions are scaled consistently, this method maintains the best possible trade-off between performance and size. The scaling equations for width, depth, and resolution are as follows:

$$\text{Width: } w = \beta^\phi$$

$$\text{Depth: } d = \alpha^\phi$$

$$\text{Resolution: } r = \gamma^\phi$$

where α , β , and γ are constants determined through empirical search, guaranteeing the best balance for the specific task at hand.

The input layer is where the data is received. Convolutional layers in EfficientNet B0 are responsible for extracting features from the input images. These convolutional layers convolve over the input pictures using filters and kernels to extract information at various spatial scales. After convolutional layers, batch normalization layers are frequently employed to normalize the activations. The network may learn intricate patterns and relationships in the data thanks to the activation layers, which

add non-linearities into the system (r20). Global average pooling is a layer that EfficientNet incorporates after the convolutional layers. By averaging each feature map, this layer compiles feature maps. There may be one or more layers that are connected after that. The output layer generates the model's final predictions in the end. This layer usually comprises ReLU or softmax activation units proportional to the number of classes in the dataset for classification problems [Barman et al. \(2024\)](#).

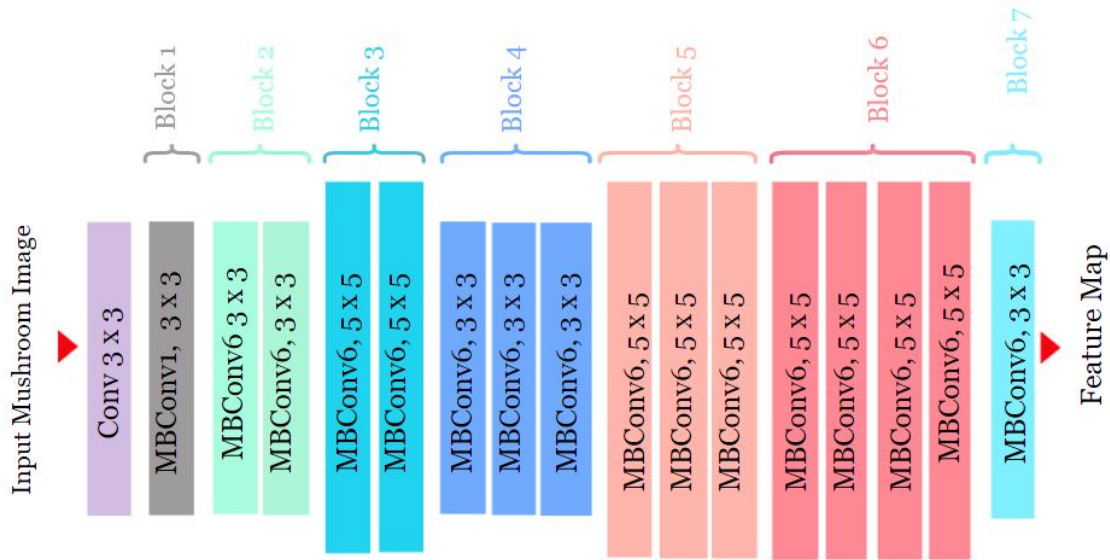


Figure 3.5: EfficientnetB0 Model Architecture

3.12.2 EfficientNetB0 for Feature Extraction

EfficientNetB0 is a convolutional neural network architecture optimized for both accuracy and efficiency. It uses a sequence of convolutional layers to interpret input images in the context of feature extraction, capturing hierarchical information at various sizes, ranging from basic edges to intricate patterns. Compared to conventional CNNs, EfficientNetB0 can extract rich and detailed features with fewer parameters thanks to its compound scaling technique, which uniformly scales the network's depth, width, and resolution. The most important features for classifying mushrooms are automatically extracted from the photos using EfficientNetB0.

3.13 Hybrid Model

A hybrid model combines two or more machine learning or deep learning techniques to leverage the strengths of each. In this project, the hybrid model integrates EfficientNetB0 and Support Vector Machine (SVM) for mushroom classification. The feature extractor, EfficientNetB0, automatically learns and extracts relevant features from the input photos. The SVM, which serves as the classifier and uses the patterns it has learnt to distinguish between edible and toxic mushrooms, is then fed these extracted characteristics. By combining the deep learning characteristics of EfficientNetB0—such as effective feature extraction—with the reliable classification performance of SVM, this combination enables the model to produce a highly accurate and effective solution for image-based classification applications.

3.13.1 SVM Model

Support Vector Machine (SVM) is a supervised machine learning algorithm widely used for classification tasks, particularly in scenarios with clear margins of separation between classes. Finding the ideal hyperplane to split a dataset into discrete groups while maximizing the margin between them is the fundamental notion behind support vector machines (SVM). Support vectors, a subset of training data points that are essential for determining the border, are what determine this hyperplane. By employing kernel functions, such as polynomial or radial basis function (RBF), to transform the input space into higher dimensions and handle both linear and non-linear classification, SVM may separate classes that are not linearly separable. SVM is a well-liked option in machine learning because of its resistance to overfitting, particularly in high-dimensional spaces. As a result, it works well for a variety of applications, such as text classification, picture recognition, and bioinformatics.

3.13.2 Transfer Learning

A machine learning technique called transfer learning involves using a model that has been trained on a sizable dataset for a task that is unrelated to it. We used our mushroom classification dataset to refine a pre-trained Efficient Net B0 model through transfer learning. High-level feature representations acquired from

a variety of images were effectively leveraged by the model by starting it with weights that had already been pre-trained on the Image Net dataset. To fine-tune the model to our particular goal, we updated the settings of the final layers while keeping the insights from Image Net. This can be expressed mathematically by modifying the model's parameters in order to minimize the loss function L .

3.13.3 Feature Extraction Process

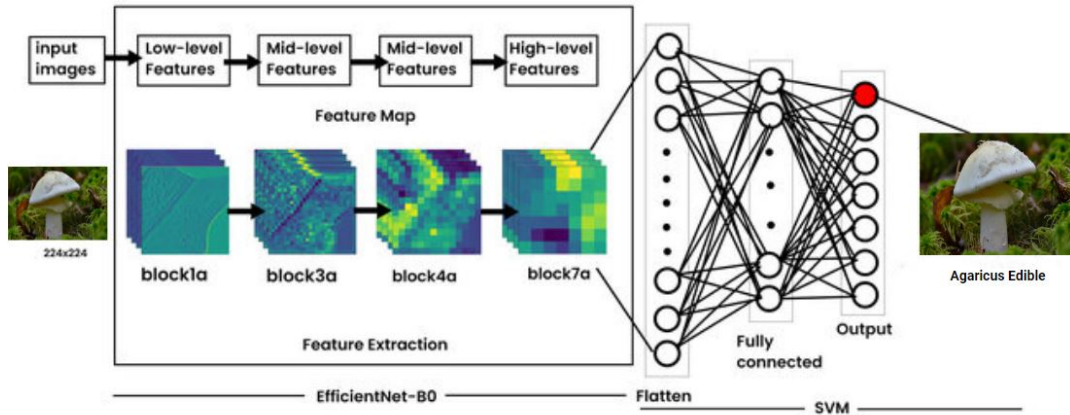


Figure 3.6: Hybrid Extraction Process

This hybrid model automatically extracts relevant features from the input mushroom images by utilizing EfficientNetB0 in the feature extraction process [Barman et al. \(2024\)](#). As images pass through the network, they are processed in various stages:

- The first layers (e.g., block1a) contain low-level features like edges and textures, whereas mid-level (block 3a, block4a) and high-level (block 7a) features gradually capture more complicated patterns including forms, colors, and species-specific structures.
- The SVM classifier receives these feature maps, which reflect the image's most important attributes, after they have been compressed into a 1D vector.

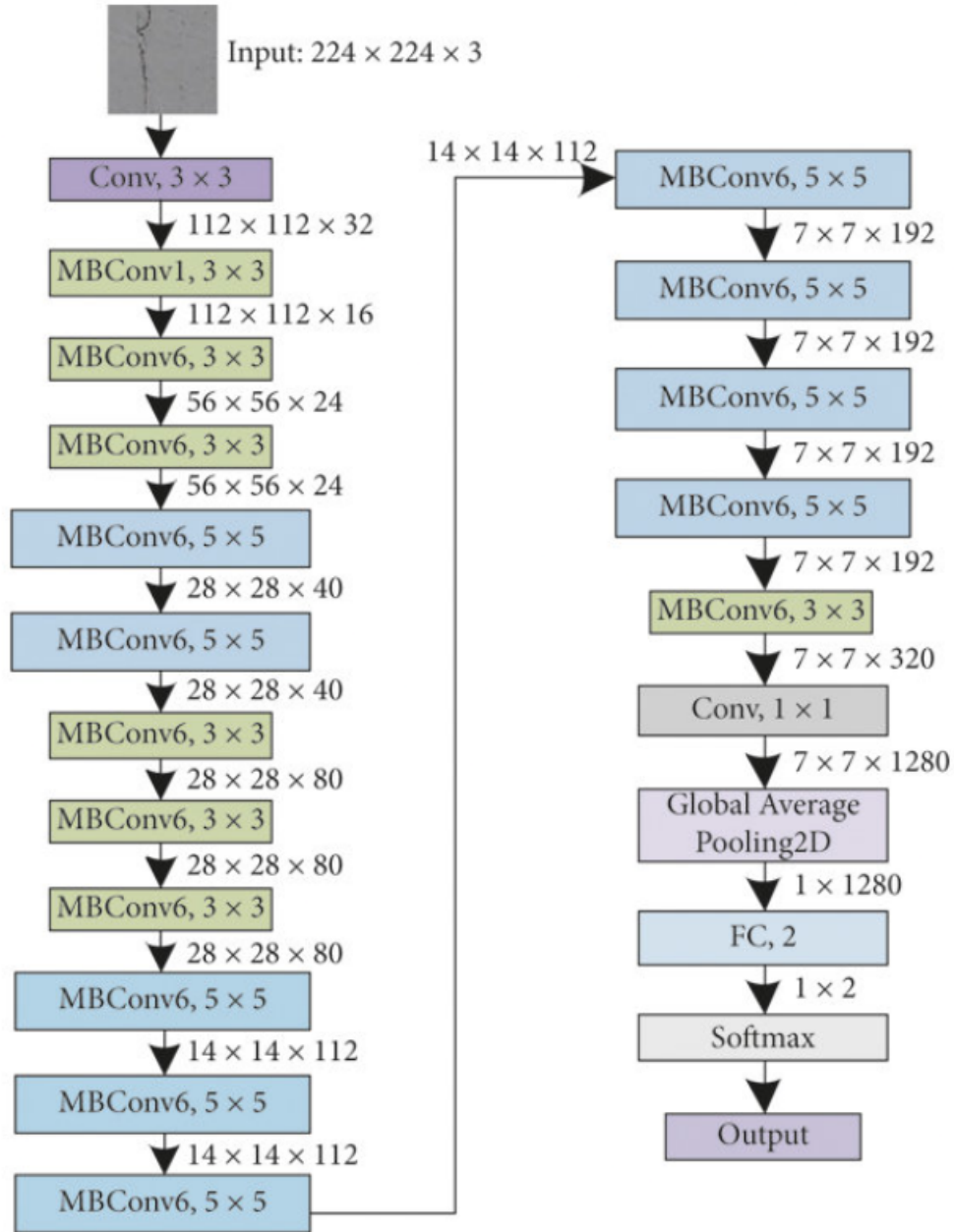


Figure 3.7: Hybrid Model Structure Diagram

- Based on its learnt patterns, SVM uses these extracted features to classify the mushrooms into categories, such as edible or toxic.

- The accuracy of the classification task is improved and the requirement for human feature engineering has been reduced due to this effective feature extraction procedure.

3.13.4 Model Training

During the model training phase, we applied numerous tactics to optimize performance. To prevent overfitting, we gave EfficientNetB0 a fixed learning rate and kept an eye on how well the model performed on the validation dataset. To prevent the model from over-training on the data, early stopping was introduced, which stopped training if the validation loss did not improve after a predetermined number of epochs. In order to lessen overfitting and enhance generalization, we also experimented with various dropout rates, choosing the one that produced the optimum balance between training and validation accuracy.

Batch Size and Epochs:

To maximize training, we experimented with various batch sizes and epochs. Following testing of several configurations, 64 was chosen as the batch size because it provided an effective trade-off between model convergence and memory consumption. The model learnt complex patterns from the mushroom dataset over the course of 50 epochs of training, which ensured adequate training without going overboard with training time. With this configuration, the model had more chances to identify intricate details in the data and produce more accurate classification outcomes.

3.13.5 Key Features of Support Vector Machine (SVM) for Classification

Support Vector Machine (SVM) is a powerful machine learning algorithm used for classification tasks. Here is some key features of Support Vector Machine (SVM):

- **Maximal Margin Classifier:** In order to improve generalization, SVM locates the hyperplane that maximizes the margin between various classes.
- **Support Vectors:** Support vectors, or the data points nearest to the hyperplane, are what the algorithm uses to directly affect the decision boundary.

- **Linear and Non-linear Classification:** Through kernel functions, SVM can handle both linearly and non-linearly separable data, providing classification tasks flexibility.
- **Kernel Trick:** SVM uses kernels like linear, polynomial, and radial basis function (RBF) to translate data into higher-dimensional spaces for improved separation.
- **Effective in High Dimensions:** When there are more dimensions than data points, SVM continues to function well in high-dimensional spaces.
- **Robustness to Overfitting:** SVM is less likely to overfit since it concentrates on the support vectors, particularly when the dataset is small but the feature space is big.
- **Versatility:** By changing the algorithm to classify multiple classes, SVM can be used for binary and multi-class classification.

3.14 Reason Behind Applying Hybrid Model

As we have gone through numerous literature reviews, we observed a few of the models that they have used and implemented in our project. However, we faced a few problems such as the accuracy was not up to the mark, some of the models took much longer to compile, and so on. Therefore, we decided to implement a hybrid model. The models used in the presented research, such as SVM and EfficientNetB0, provide clear benefits for the mushroom classification work. With its expandable architecture, EfficientNetB0 allows for high accuracy and efficient feature extraction. By balancing network depth, width, and resolution through the use of the compound scaling, it creates an optimized model that excels on little datasets. SVM is a useful tool for the classification stage following feature extraction since it effectively handles high-dimensional feature spaces, which is a complement to EfficientNetB0. In order to gain improved accuracy in differentiating between edible and hazardous mushrooms, this hybrid method integrates the robustness of SVM for classification with the deep learning capabilities for the representation of features.

Chapter 4

Design and Development

4.1 Project Management and Financial Activities

Our project management activities have been established to guide the development of a web application for classifying mushrooms. The goals of the project must be defined, a dedicated group has to be formed under our supervision, and duties must be clearly defined throughout the start phase. Creating a complete project plan, assigning resources, and managing risks are all included in planning. In order to build the hybrid model using EfficientNetB0 and SVMs, we first implement data collecting and preprocessing procedures during execution. Monitoring was done on a continuous basis to ensure that risks were reduced, data quality standards were upheld, and project development was in line with schedules. Evaluating the model's performance and recording the most important lessons are part of the closure step.

Financial activities have been documented according to a comprehensive spending plan that accounts for personnel, equipment, and other expenses. We carefully monitor expenses, allocate resources according to the most important project phases, and provide stakeholders with regular financial updates. Our methodology ensures efficient project administration and fiscal responsibility, steering the effective creation of the web-based application for mushroom classification.

4.1.1 Financial Report Project Budget

Table 4.1: Budget Summary

Discretion	Budget
Hardware Costs	4,23,000
Transportation Cost	3,000
Miscellaneous Cost	1,500

Table 4.2: Total cost of Hardware Costs

Hardware Description	Unit Price	Quantity	Amount
CPU Cooler Cooler Master MasterLiquid L360L	13,000	1	13,000
GIGABYTE GeForce RTX 3080 Ti VISION OC 12GB GDDR6X	143,000	1	143,000
Hard Disk Drive Seagate Barracuda 2TB	5,400	1	5,400
Monitor Asus TUF Gaming VG27AQ1A 27 inch 2K	52,000	1	52,000
Motherboard Gigabyte Z590 VISION G 11th Gen	24,000	1	24,000
Power Supply Corsair RM650 650Watt 80 Plus Gold	9,000	1	9,000
Processor Intel 11th Gen Rocket Lake Core i9 11900K 3.50-5.20GHz,	56,000	1	56,000
RAM Gigabyte AORUS RGB 16GB (2x8GB) DDR4 4400MHz	7,800	2	15,600
Samsung Galaxy s24 ultra	1,05,000	1	1,05,000
Total Budget			4,23,000

4.2 Adopted Tools and techniques

4.2.1 Hardware Requirements

The system demands the following hardware components to operate optimally:

- Processor: Intel 12th Gen Core i9-11900K Alder Lake Processor

- RAM: 32 GB Ram
- Storage: 12 TB
- Nvidia GeForce RTX 3080 Ti OC 12GB GDDR6X Graphics Card
- Display: Monitor or screen for accessing the user interface
- Input devices: Keyboard and mouse or equivalent input methods

4.2.2 Software Requirements

The following software components are necessary for the proper functioning of the system:

- Python 3.x: The system requires Python version 3.x for development and execution.
- Machine Learning Libraries: Including Scikit-learn, TensorFlow, and Tensor-Flow for implementing machine learning models.
- NVIDIA CUDA for train the model faster
- Version Control System: Utilize Git to oversee the source code and promote collaboration within the development team.
- Integrated Development Environment (IDE): Utilize tools like Anaconda and Visual Studio Code to streamline coding and debugging processes for increased efficiency.

4.3 Design the Solution

1. Concept Generation: The classification of mushrooms is a topic that has numerous conceptual solutions. The main idea was built on deep learning models that classify images using data from images. Three primary strategies were developed:

- Convolutional Neural Network (CNN): A classic CNN architecture for classifying mushrooms directly.

- Transfer Learning with VGG16: Fully connected layers for classification and a pre-trained VGG16 model for feature extraction.
- Hybrid EfficientNetB0 + SVM Model: To balance computational efficiency and accuracy, SVM was utilized for classification and EfficientNetB0 for feature extraction.

A concept tree diagram that illustrated the phases of feature extraction and classification methods was used to illustrate each strategy.

2. Concept Reduction: A procedure for making decisions was used to choose the optimal course of action. The solutions were assessed using the subsequent standards:

- Accuracy: The ability to categorize mushrooms as poisonous or edible.
- Computational Efficiency: Processing power and memory consumption are needed.
- Generalization: Evaluate performance on unknown data and prevent overfitting.
- Ease of Implementation: Deployment and integration simplicity.

It was decided to use the Hybrid EfficientNetB0 + SVM model after weighing the available possibilities. Because of its reduced accuracy, the standard CNN was rejected, and VGG16 was found to be computationally more expensive than EfficientNetB0. Performance and efficiency were balanced by EfficientNetB0, while robust classification was supplied by SVM.

3. Justification and Novelty: The solution is new in that it optimizes accuracy while lowering computing complexity for mushroom classification in a resource-constrained environment by integrating the feature extraction of EfficientNetB0 with the classification of SVM.

4.4 Implementation

4.4.1 Import necessary Libraries and Modules

```
1 import os
2 import cv2
3 import joblib
4 import random
5 import numpy as np
6 import pandas as pd
7 import seaborn as sns
8 import tensorflow as tf
9 import matplotlib.pyplot as plt
10 import efficientnet.tfkeras as efn
```

This code contains the initial setup, which imports the libraries and modules required for our project on classifying mushrooms. Here's a breakdown of what each import does:

- `import os` : Provides a mechanism to interface with the operating system. The reading and writing of files, as well as other file and directory activities, are handled by this module.
- `import cv2`: A well-liked library for computer vision applications, good for image processing jobs like data reading, augmentation, and scaling.
- `import joblib`: A library for storing and loading massive datasets, such as machine learning models. When working with huge numpy arrays, it is usually faster than pickle.
- `import random`: It offers utilities for shuffling data and creating random numbers, which are helpful for reproducibly generating random seeds or randomizing data partitions.
- `import numpy as np`: A core Python library for numerical computation that is used to manage arrays and perform matrix operations—two tasks that are critical to workflows in deep learning and machine learning.
- `import pandas as pd`: A library for data manipulation that can be used to load and manage datasets, especially those in tabular form (CSV, Excel, etc.).

- import seaborn as sns: A matplotlib-based statistical data visualization package that is helpful for producing eye-catching and educational statistical visualizations.
- import tensorflow as tf: A widely used deep learning package that facilitates the creation and training of models, like EfficientNet. It will be heavily utilized for model validation and training.
- import matplotlib.pyplot as plt: A plotting library for creating scatter plots, line charts, and histograms, among other visualizations.
- import efficientnet.tfkeras as efn : An additional module that lets us utilize the TensorFlow Keras framework with EfficientNet models. It appears from this import that EfficientNet is a component of our deep learning model.

```

1 from PIL import Image
2 from tqdm import tqdm
3 from skimage import io
4 from sklearn.svm import SVC
5 from sklearn.manifold import TSNE
6 from tensorflow.keras.models import Model
7 from keras.preprocessing.image import ImageDataGenerator
8 from sklearn.metrics import precision_score, recall_score,
9 f1_score, confusion_matrix

```

These libraries are essential at different points in our project:

- PIL and skimage.io for image preprocessing,
- SVM for classification,
- tqdm for progress tracking,
- ImageDataGenerator for data augmentation,
- Metrics for performance evaluation.

4.4.2 Data Processing

```
1 DATA_DIR = 'Data'
2 TRAIN_DATA_DIR = os.path.join(DATA_DIR, 'train')
3 TEST_DATA_DIR = os.path.join(DATA_DIR, 'valid')
```

The base directory path, which contains the dataset, is stored in the `DATA_DIR` variable, which is defined in this code. The `TRAIN_DATA_DIR` and `TEST_DATA_DIR` variables are formed by attaching the subdirectory names ('train' and 'valid', respectively) to the base data directory path using Python's `os.path.join()` function. This method ensures that the code is portable across various contexts without hard-coding absolute paths by managing the file directories for the training and testing (validation) datasets in a clean and dynamic manner.

```
1 from tqdm import tqdm
2 training_data = []
3
4 for c in CATEGORIES:
5     path = os.path.join(TRAIN_DATA_DIR, c)
6     class_num = CATEGORIES.index(c)
7     for img in tqdm(os.listdir(path)):
8         try:
9             img_array = cv2.imread(os.path.join(path, img))
10            img_resized = cv2.resize(img_array,
11                                     (IMG_SIZE, IMG_SIZE))
12            training_data.append([img_resized, class_num])
13        except Exception as e:
14            pass
15
16 print(len(training_data))
```

To load and prepare images for machine learning model training, use this bit of code. This is a detailed breakdown of the procedure:

Initialize training_data: In order to hold the processed picture data and the associated class labels, an empty list called `training_data` is launched.

Loop Through Categories (CATEGORIES): For each category ("edible" or "toxic" in our mushroom dataset), the loop iterates through the `CATEGORIES` list.

- `path`: It connects the current category (`c`) to the path of the training data directory (`TRAIN_DATA_DIR`).

- `class_num`: The class number, which is the category index, is used to assign a numerical label to each image (0 meaning "edible" and 1 meaning "toxic").

Loop Through Images (`tqdm(os.listdir(path))`): For each image in the category directory:

- Load the image: The image is read from the supplied file directory using `cv2.imread()`.
- Resize the image: Using `cv2.resize()`, the image is resized to a standard size (specified by `IMG_SIZE`). This assures that, crucially for models input, every image in the dataset has the same dimensions.
- Store the image and label: The resized image is added to `training_data` along with its class label (`class_num`).

Error Handling (`try-except`): Using a `try-except` block, all errors that occur when reading or processing an image are skipped.

Display the number of processed images: `Print(len(training_data))` shows the overall amount of loaded and preprocessed images after all the images have been processed.

4.4.3 Model

```
1  enet = efn.EfficientNetB0(  
2      input_shape=(224, 224, 3),  
3      weights='imagenet',  
4      include_top=False  
5  )  
6  
7  enet.trainable = False  
8  x = enet.output  
9  x = tf.keras.layers.GlobalMaxPooling2D()(x)  
10 x = tf.keras.layers.Dense(256, activation='relu')(x)  
11 x = tf.keras.layers.BatchNormalization()(x)  
12 x = tf.keras.layers.Dropout(0.5)(x)  
13 y = tf.keras.layers.Dense(classes, activation='softmax')(x)  
14  
15 e_model_b0= tf.keras.Model(inputs=enet.input, outputs=y)  
16 e_model_b0.compile(  
17     optimizer = tf.keras.optimizers.Adam(learning_rate=5e-4),  
18     loss='sparse_categorical_crossentropy',  
19     metrics=['accuracy']  
20 )
```

This code creates and assembles a unique deep learning model for image classification that is based on EfficientNetB0. Here's a breakdown of the essential steps:

Load Pretrained EfficientNetB0 Model ('enet'):

- For RGB images, a specific input shape of (224, 224, 3) is used to initialize EfficientNetB0.
- `weights='imagenet'`: The ImageNet dataset's pretrained weights are used in this, which can enhance training efficiency on new tasks.
- `include_top=False`: The final classification layer is not included here since we will add new layers customized to our particular classification requirement.
- `enet.trainable = False`: By doing this, the pretrained features are kept because the EfficientNetB0 base is frozen, preventing weight updates during training.

Add Custom Layers:

- `GlobalMaxPooling2D()`: This layer applies global max pooling to decrease the spatial dimensions of the feature map while producing the most important features.

- `Dense(256, activation='relu')`: A fully connected layer with 256 units and ReLU activation is implemented for learning complicated patterns from the collected features.
- `BatchNormalization()`: By normalizing the dense layer's output, this speeds up and stabilizes training.
- `Dropout(0.5)`: 50% of the cells are randomly deactivated during training by this dropout layer to avoid overfitting.
- `Dense(classes, activation='softmax')`: The multi-class classification output from the final layer including 'softmax' activation. The number of target classes and the number of units (classes) must match.

Create the Model ('e_model.b0'): The EfficientNetB0 input and the output from the custom layers are specified in order to generate the model.

Compile the Model:

- `optimizer=tf.keras.optimizers.Adam(learning_rate=5e-4)`: An Adam optimiser with a learning rate of 0.0005 has been used.
- `loss='sparse_categorical_crossentropy'`: Sparse categorical cross-entropy is the loss function that is employed, and it works well for classification of multiple classes with integer labels.
- `metrics=['accuracy']`: During training and validation, the accuracy measure is utilized to assess the performance of the model.

This model architecture integrates specially designed dense layers for the classification problem with a potent pre-trained EfficientNetB0 for feature extraction. The model learns additional features for distinguishing between toxic and edible mushrooms quickly by freezing the underlying model and concentrating training on the custom layers.

```

1 features_train = e_model_b0.predict(X_train)
2 features_test = e_model_b0.predict(X_test)
3
4 features_train_flat = features_train.reshape
5 (features_train.shape[0], -1)
6 features_test_flat = features_test.reshape
7 (features_test.shape[0], -1)
8
9 svm = SVC(kernel='rbf', C=1.0, gamma='auto')
10 svm.fit(features_train_flat, Y_train)

```

This snippet of code integrates a Support Vector Machine (SVM) classifier with the feature extraction capabilities of the EfficientNetB0 model in a hybrid approach. Here is a detailed explanation:

Feature Extraction Using EfficientNetB0:

`features_train = e_model_b0.predict(X_train)` and `features_test = e_model_b0.predict(X_test)`:

- The training (`X_train`) and testing (`X_test`) datasets are utilized to extract features using the least EfficientNetB0 model (`e_model_b0`). These characteristics indicate higher-level patterns that the model has learnt, as opposed to classifying the raw image data directly.
- Since this method uses derived features rather than raw pixel data, it drastically lowers the complexity of the input to the SVM.

Flatten the Features:

`features_train_flat = features_train.reshape(features_train.shape[0], -1)` and `features_test_flat = features_test.reshape(features_test.shape[0], -1)`:

- 2D or 3D tensors are the features that are extracted. The features are transformed into 2D arrays using `reshape()` in order to make them compatible with the SVM, which needs 1D input. A single vector is created by flattening each feature map.
- A single feature vector per image is contained in the resulting arrays, `features_train_flat` and `features_test_flat`.

Initialize and Train the SVM:

`svm = SVC(kernel=rbf, C=1.0, gamma='auto'):`

- The radial basis function (RBF) kernel of an SVM classifier is initialized. Since the RBF kernel transfers the data into a higher-dimensional space, where separating hyperplanes are easier to identify, it is a good choice for non-linear classification applications.
- The regularization parameter, $C=1.0$, controls how much of a trade-off is made between getting a low testing error and low training error.
- The range of impact that a single training example can have is defined by `gamma='auto'`. It is set to $1/n_{\text{features}}$ with `gamma='auto'`, where `n_features` is the total number of features in the input.

`svm.fit(features_train_flat, Y_train):`

The vectors of features that have been flattened (`features_train_flat`) and the labels that relate to them (`Y_train`) are used to train the SVM. It learns to classify the feature vectors generated by EfficientNetB0 into distinct categories (edible or toxic mushrooms, for example).

Strong classification accuracy can be achieved with this hybrid strategy, particularly in situations where combining deep learning with conventional machine learning techniques is advantageous.

4.5 Evaluate the solution

4.5.1 Evaluation Metrics

Evaluation metrics act as essential instruments for evaluating a machine learning model's performance. Common metrics are recall, which shows the percentage of true positives in the data that are actually positives, precision, which shows the percentage of true positive predictions out of all positive predictions made by the model, and accuracy, which measures the percentage of correct predictions. The F1-score is a single metric that performs well with unbalanced datasets because it strikes a compromise between precision and recall. The model's performance

across all classes is visualized with the use of the confusion matrix, which provides a thorough breakdown of true positives, false positives, true negatives, and false negatives. These measures illuminate how well a model applies to previously untested data.

4.5.2 Precision Score

A measure used in classification tasks to assess the precision of positive predictions is the precision score. Its definition is the ratio of genuine positives to the overall number of positive predictions (false and true) that the model generates.

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive}$$

4.5.3 Recall Score

In classification problems, the recall score is a metric that assesses the model's accuracy in identifying all pertinent positive cases. The ratio of true positives, or positively anticipated cases, to the total number of actual positives, or true positives plus false negatives, is how it is defined. A high recall score means that the model is good at detecting the majority of positive cases, which is important when there are major repercussions for missing a positive instance (false negative).

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative}$$

4.5.4 F1 score

Particularly when working with unbalanced datasets, the F1 score is a statistic that provides a fair assessment of a model's performance by combining precision and recall.

$$F1Score = 2 * \frac{PrecisionRecall}{Precision + Recall}$$

It is the harmonic mean of recall and precision, making sure that they are given equal weight. Better performance is indicated by an F1 score that is closer to 1. The score goes from 0 to 1. It is especially helpful in situations where recall and precision must be traded off because it evaluates the model's ability to manage both erroneous positives and false negatives. When accurate positive predictions are crucial, the F1 score is a useful indicator; but, it can also be expensive to make too many false positives (poor precision) or overlook actual positives (low recall).

4.5.5 Accuracy Score

The accuracy score, which measures the percentage of accurate predictions a model makes, is a popular evaluation metric used in classification projects. It is computed as the ratio of the total number of forecasts to the number of accurate predictions (including true negatives and positives).

$$F1Score = \frac{TP + TN}{TP + TN + FP + FN}$$

4.5.6 Confusion Matrix

A confusion matrix is a visual tool for performance evaluation that shows the outcomes of a model's predictions in classification tasks. The efficiency of the classification model can be understood by referring to the confusion matrix.

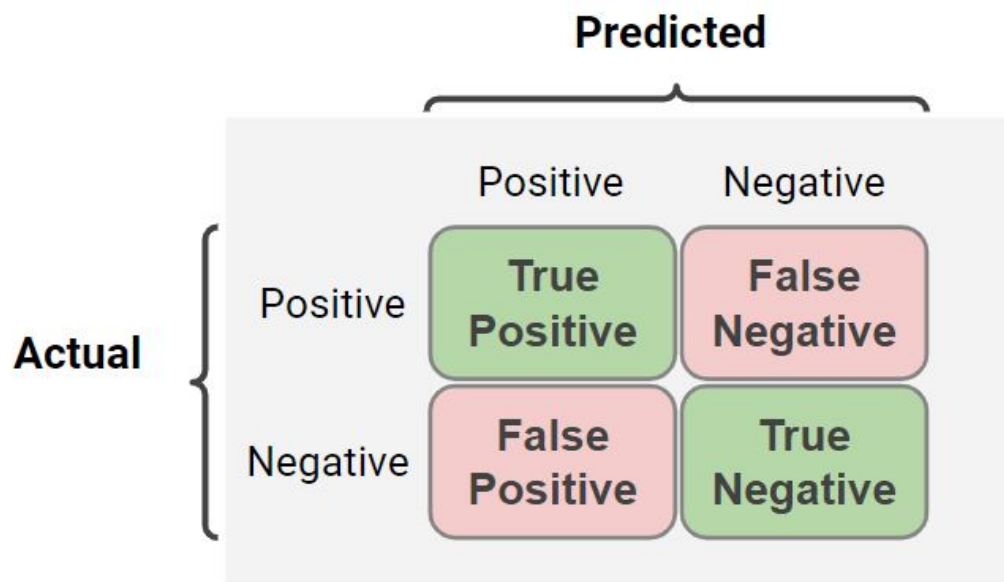


Figure 4.1: Confusion Matrix

Chapter 5

Result analysis

5.1 Model Performance

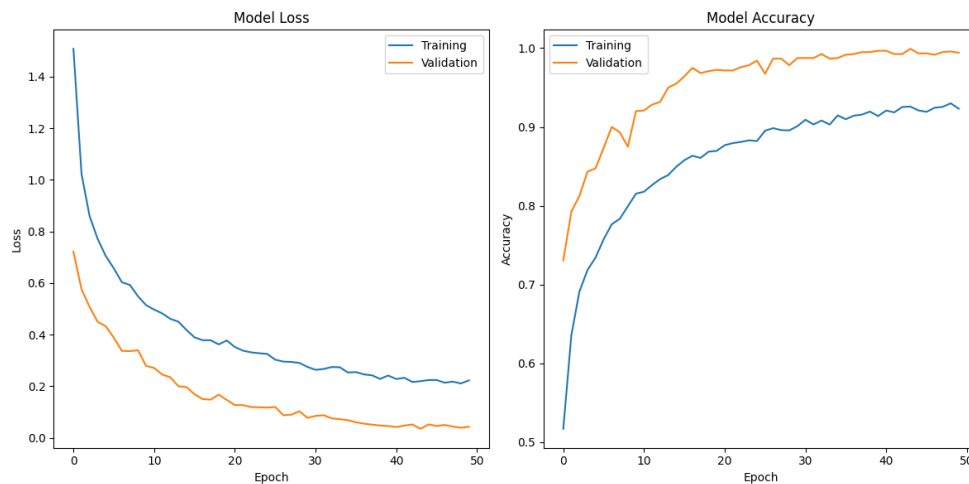


Figure 5.1: Hybrid Plot training and validation loss values and training and validation accuracy values

Strong performance characteristics are shown by the hybrid model that combines a Support Vector Machine (SVM) classifier with EfficientNetB0 as a feature extractor. The model's accuracy and loss over 50 epochs of training and validation are displayed in the graphs:

- **Model Loss (Left Graph):** The model is learning to minimize the error on the training dataset as shown by the training loss, which is represented by

the blue line. It begins high and gradually drops. The model exhibits strong generalization to previously encountered data as evidenced by the decreasing validation loss (orange line). Little difference exists between training and validation loss, indicating no discernible overfitting.

- **Model Accuracy (Right Graph):** The model's ability to identify the correct class for training data is improving as seen by the training accuracy (blue line), which begins low and rises steadily to over 85%. The model appears to perform better on the validation set, as indicated by the validation accuracy (orange line), which starts higher than the training accuracy and reaches almost 95%. Good model generalization is indicated by the training and validation accuracy closely aligning. Overall, both graphs' patterns point to a strong model that well balances training and validation performance.

5.2 Class-wise Evaluation

Precision, Recall, and F1-Score: These metrics offer a thorough picture of how well the model performs for certain classes. We computed the following metrics by averaging the scores of all classes:

Hybrid Classification Report (Train)				
	precision	recall	f1-score	support
Agaricus_Bisporus_Edible	0.9950000	0.9950000	0.9950000	800
Agaricus_Edible	0.9949495	0.9850000	0.9899497	800
Amanita_Toxic	1.0000000	1.0000000	1.0000000	800
Calocybe_Indica_Edible	0.9852217	1.0000000	0.9925558	800
Deathcap_Toxic	0.9900498	0.9950000	0.9925187	800
Oyster_Edible	1.0000000	0.9900000	0.9949749	800
accuracy			0.9941667	4800
macro avg	0.9942035	0.9941667	0.9941665	4800
weighted avg	0.9942035	0.9941667	0.9941665	4800

Table 5.1: Hybrid Classification Report (Train)

Hybrid Classification Report (Test)				
	precision	recall	f1-score	support
Agaricus_Bisporus_Edible	0.9950000	0.9950000	0.9950000	200
Agaricus_Edible	0.9949749	0.9900000	0.9924812	200
Amanita_Toxic	1.0000000	1.0000000	1.0000000	200
Calocybe_Indica_Edible	0.9900990	1.0000000	0.9950249	200
Deathcap_Toxic	0.9900498	0.9950000	0.9925187	200
Oyster_Edible	1.0000000	0.9900000	0.9949749	200
accuracy			0.9950000	1200
macro avg	0.9950206	0.9950000	0.9949999	1200
weighted avg	0.9950206	0.9950000	0.9949999	1200

Table 5.2: Hybrid Classification Report (Test)

Observation: For the testing dataset, the hybrid model produced a better result with an accuracy of 99.50%.

5.3 Confusion Matrix

The confusion matrix compares the actual and expected classifications to assess how well a classification model performs. A more thorough examination of the model's accuracy and performance across several classes is made possible by this matrix, which assists in locating the model's error spots.

- Agaricus Edible 0
- Agaricus Bisporus Edible 1
- Calocybe Indica Edible 2
- Oyster Edible 3
- Amanita Toxic 4
- Deathcap Toxic 5

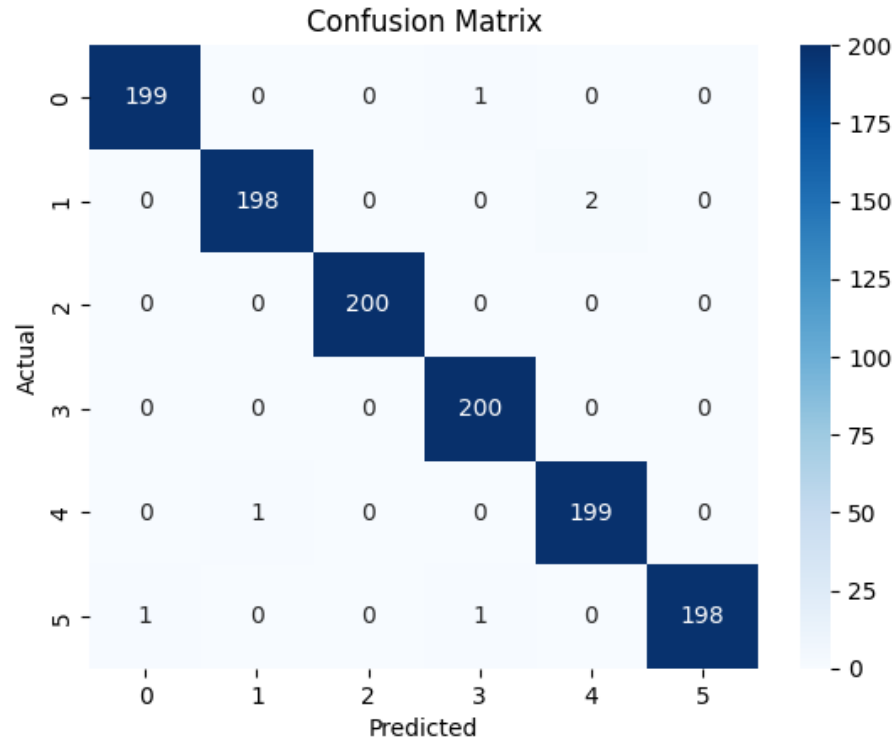


Figure 5.2: Hybrid Confusion Matrix

The classification results for the six classes (0–5), each with 200 test samples, are displayed in the confusion matrix. The following matrix provides illumination on how well the model performed using the test data:

- **Diagonal values (correct classifications):** These show how many samples for each class were accurately classified:
 - Class 0: 199 samples correctly classified out of 200.
 - Class 1: 198 samples correctly classified.
 - Class 2: 200 samples correctly classified (perfect accuracy).
 - Class 3: 200 samples correctly classified (perfect accuracy).
 - Class 4: 199 samples correctly classified.
 - Class 5: 198 samples correctly classified.
- **Off-diagonal values (misclassifications):** These represent the errors in prediction:

- For class 0, one sample was misclassified as class 3.
- For class 1, two samples were misclassified as class 4.
- For class 4, one sample was misclassified as class 1.
- For class 5, two misclassifications occurred: one as class 0 and one as class 3.

The matrix shows that the model functioned remarkably well, showing high accuracy across all classes and very few misclassifications.

5.4 Comparison of Model Performance

Model	Number of Images	Number of Classes	Accuracy	Year
CNN	8556	11	74.00%	2021
VGG 16	11000	18	84.96%	2023
ResNet 50	8190	45	88.40%	2021
Mobilenet V2	13587	27	93.10%	2021
inception v3	600	6	92.70%	2022
Efficientnet B0	5500	8	86.60%	2023
Naive Bayes	900	9	71.50%	2022
Decision Tree	8000	22	98.82%	2022

Table 5.3: Comparative analysis of research

Model	Number of Images	Number of Classes	Accuracy	Year
CNN	6000	6	86.00%	2024
VGG 16	6000	6	87.93%	2024
ResNet 50	6000	6	91.37%	2024
Hybrid	6000	6	99.50%	2024

Table 5.4: Comparative analysis of our own implement model

Observation: The tables presents a comparative analysis of different deep learning models, that we found in different research paper and we implemented including CNN, VGG16, ResNet50, MobileNetV2, InceptionV3, Naive Bayes, Decision Tree, EfficientNetB0, and a Hybrid model, based on the number of images, number of classes, accuracy, and year of development.

- **CNN:** CNN model attained a 86% accuracy rate after being trained with 6000 images in 6 classes.
- **VGG16:** VGG16 model achieved an accuracy of 87.93% using 6000 images in 6 classes.
- **ResNet50:** ResNet50 model achieved 91.37% accuracy with 6000 images and 6 classes.
- **Hybrid:** Training on 4800 images and testing on 1200 images in total 6000 images in 6 classes, the hybrid model used an SVM for classification and EfficientNetB0 for feature extraction. This approach is strong and successful, as evidenced by the excellent accuracy of 99.50% that it attained.

Chapter 6

Conclusions

6.1 Societal, health, safety, legal and cultural aspects

Deep learning-based mushroom classification system in Bangladesh has the potential to affect legal, cultural, sociological, health, and safety concerns. These problems are brought about by the automation, real-time prediction capabilities, and data collection made possible by machine learning algorithms. An assessment of each factor and a suggested plan of action to allay any worries are provided below.

- **Societal Impact:** As the risk of mushroom poisoning is reduced, the technique can have a positive impact on society. It helps promote safe consumption practices by helping foragers, farmers, and consumers distinguish between edible and hazardous mushrooms with reliability.
- **Health and Safety Issue:** In order to address health and safety concerns, the system recognises poisonous mushroom species that, if consumed, could result in fatalities or major diseases. It can assist medical personnel in situations where mushroom poisoning is suspected and help prevent unintentional poisoning by accurately classifying foods.
- **Legal Issue:** Legally speaking, in areas where wild mushroom collecting is prevalent, employing the method may assist in adhering to food safety rules and laws. It can assist authorities in implementing regulations pertaining to the gathering and selling of edible mushrooms, guaranteeing that only vetted and safe types are dispersed.

- **Cultural Issue:** In many cultures, mushrooms are significant cultural objects. Certain species are utilized in regional cuisine, traditional remedies, or religious ceremonies. In addition to encouraging sustainable gathering methods, the system can help communities maintain the safe use of mushrooms as part of their cultural history.

The mushroom classification system seeks to assist legal compliance, uphold cultural traditions, and improve public safety by attending to these factors.

6.2 Impact on Environment and Sustainability

A research project that uses deep learning to classify edible and toxic mushrooms has potential benefits for sustainability and the environment. The project minimizes agricultural strain and promotes sustainable foraging behaviors by encouraging the safe identification of edible mushrooms, hence decreasing the need for farmed mushrooms. Proper identification can help avoid the overharvesting of specific mushroom species, thereby preserving biodiversity. Furthermore, by prohibiting the harvesting of harmful or endangered species that are essential to the functioning of forest ecosystems, the tool lowers the possibility of environmental harm. From a sustainability perspective, the project promotes a healthy relationship with the environment by encouraging reliance on natural, renewable food sources. But in order to guarantee sustainability over the long run, the solution needs to raise user awareness of conservation and ethical foraging techniques.

6.3 Ethical and professional principles

A strict commitment to professional ethics and ethical standards was prioritized during the creation of the mushroom classification project, especially with regard to public and environmental safety. Since misidentifying toxic mushrooms might pose health risks, Design for Safety techniques were integrated into the design process to reduce potential hazards. Extensive testing and validation procedures were used on a variety of datasets to provide the highest level of accuracy in distinguishing between edible and toxic mushrooms in order to incorporate safety. The model was created with the intention of preventing the unexpected effects of misclassification.

It emphasizes transparency on its limitations and offers clear, practical guidance, encouraging users to consult an expert when in doubt.

To find potential points of failure, such as inaccurate classifications because of low image quality or rare species not included in the dataset, a thorough Hazards and Failure Analysis was carried out. By adding to the dataset, using strong preprocessing methods, and including a confidence threshold in the model's predictions to notify users when the model is uncertain about its classification, these risks were reduced. Additionally, the system logs predictions for later research and ongoing enhancement.

Legally speaking, the project conforms to national and international food safety laws by making sure it doesn't encourage the ingestion of wild mushrooms that haven't been confirmed. The tool is intended to be used in conjunction with existing restrictions concerning mushroom consumption and foraging, not as a primary means of making decisions. Furthermore, we maintained our ethical obligation to the public and the environment by making sure that no sensitive or personal data was captured during system use. This also addressed privacy and data security concerns.

6.4 Brief Summary

The development of a trustworthy and accurate deep learning-based system that can successfully classify mushrooms as either edible or toxic, particularly in the setting of Bangladesh, is one of the project's anticipated goals. The system's goal is to significantly reduce the hazards involved with mushroom foraging by giving users—consumers and foragers alike—a tool to recognise potentially dangerous species. EfficientNetB0 and SVM are combined in the model's construction to provide excellent accuracy and good generalization over a variety of mushroom species.

Among the notable achievements are:

- The effective development of an edible and poisonous species labeled mushroom image dataset.

- The application of a convolutional neural network (CNN) that has been optimized to distinguish between different kinds of mushrooms for image classification.
- Including an SVM classifier to improve classification precision.
- Extensive evaluation measures to verify model performance, including accuracy, precision, recall, and F1 score.
- Easy-to-use visual aids for tracking model performance, including learning curves, confusion matrices, and example picture grids.

The project's ultimate goal is to improve public health and safety by reducing the likelihood that people would consume harmful mushrooms and encouraging ecologically friendly foraging techniques.

6.5 Future works

There are numerous ways to build on the current work in the future. First, the generalization and robustness of the model can be further improved by growing the dataset with larger and more varied samples. Furthermore, experimenting with alternative sophisticated deep learning architectures, including transformer-based models or variations of EfficientNet, may enhance computing efficiency and accuracy. Using transfer learning methods from trained models in related domains is another area to investigate. Incorporating explainable AI (XAI) techniques would also increase the model's decision-making process's transparency and understanding. Lastly, a useful expansion would be to implement the model in real-world applications, such a smartphone app for classifying mushrooms, which would provide users with an approachable tool for classifying mushrooms in the field.

References

- Aravindhana, V., Sanjana, R., Raaj, A. M., Rani, C., & Kumar, M. R. (2024). Advancing sustainability in mushroom cultivation by leveraging iot and ml techniques. In *2024 3rd international conference on artificial intelligence for internet of things (aiiot)* (pp. 1–6).
- Barman, S., Farid, F. A., Raihan, J., Khan, N. A., Hafiz, M. F. B., Bhattacharya, A., ... others (2024). Optimized crop disease identification in bangladesh: A deep learning and svm hybrid model for rice, potato, and corn. *Journal of Imaging*, 10(8), 183.
- Cengil, E., & Çınar, A. (2021). Poisonous mushroom detection using yolov5. *Turkish Journal of Science and Technology*, 16(1), 119–127.
- Demirel, Y., & Demirel, G. (2023). Deep learning based approach for classification of mushrooms. *Gazi University Journal of Science Part A: Engineering and Innovation*, 10(4), 487–498.
- Devika, G., & Karegowda, A. G. (2021). Identification of edible and non-edible mushroom through convolution neural network. In *3rd international conference on integrated intelligent computing communication & security (iciic 2021)* (pp. 312–321).
- El-Ramady, H., Abdalla, N., Badgar, K., Llanaj, X., Törös, G., Hajdú, P., ... Prokisch, J. (2022). Edible mushrooms for sustainable and healthy human food: nutritional and medicinal attributes. *Sustainability*, 14(9), 4941.
- Ferdousi, J., Riyadh, Z. A., Hossain, M. I., Saha, S. R., & Zakaria, M. (2020). Mushroom production benefits, status, challenges and opportunities in bangladesh: A review. *Annual Research & Review in Biology*, 34(6), 1–13.
- Jo, W.-S., Hossain, M. A., & Park, S.-C. (2014). Toxicological profiles of poisonous, edible, and medicinal mushrooms. *Mycobiology*, 42(3), 215–220.
- Ketwongsa, W., Boonlue, S., & Kokaew, U. (2022). A new deep learning model for the classification of poisonous and edible mushrooms based on improved alexnet convolutional neural network. *Applied Sciences*, 12(7), 3409.

- Li, W., Pires, S. M., Liu, Z., Liang, J., Wang, Y., Chen, W., . . . others (2021). Mushroom poisoning outbreaks—china, 2010–2020. *China CDC weekly*, 3(24), 518.
- Moysiadis, V., Kokkonis, G., Bibi, S., Moscholios, I., Maropoulos, N., & Sarigiannidis, P. (2023). Monitoring mushroom growth with machine learning. *Agriculture*, 13(1), 223.
- Ottom, M. A., Alawad, N. A., & Nahar, K. (2019). Classification of mushroom fungi using machine learning techniques. *International Journal of Advanced Trends in Computer Science and Engineering*, 8(5), 2378–2385.
- Rianasari, D., Triana, M. N., Dewi, M. R., Astutik, Y., et al. (2022). The classification of mushroom types using naïve bayes and principal component analysis. *JISA (Jurnal Informatika dan Sains)*, 5(2), 124–130.
- Shehab, E., Nabil, R., et al. (2023). The classification of mushroom using ml. *Kafrelsheikh Journal of Information Sciences*, 4(2).
- Tutuncu, K., Cinar, I., Kursun, R., & Koklu, M. (2022). Edible and poisonous mushrooms classification by machine learning algorithms. In *2022 11th mediterranean conference on embedded computing (meco)* (pp. 1–4).
- Wang, Y., Yang, L., Chen, H., Hussain, A., Ma, C., & Al-gabri, M. (2022). Mushroom-yolo: A deep learning algorithm for mushroom growth recognition based on improved yolov5 in agriculture 4.0. In *2022 ieee 20th international conference on industrial informatics (indin)* (pp. 239–244).
- Zahan, N., Hasan, M. Z., Malek, M. A., & Reya, S. S. (2021). A deep learning-based approach for edible, inedible and poisonous mushroom classification. In *2021 international conference on information and communication technology for sustainable development (icict4sd)* (pp. 440–444).