Uncovering The Hidden Treasures Of The Mushroom Kingdom: A Classification Analysis

1. INTRODUCTION

1.1 Project Overview

The project aims to develop a mushroom classification system using deep learning techniques. The system will analyse images of mushrooms and classify them into three categories: Boletus, Lactarius, and Russula. This classification will provide valuable information for users interested in identifying different mushroom species accurately.

1.2 Purpose

The purpose of this project is to create a robust and accurate solution for the optical recognition of mushroom species. By utilizing deep learning and transfer learning methods, the system aims to achieve high-performance classification of mushrooms based on their visual characteristics. This will assist mushroom enthusiasts, researchers, and professionals in accurately identifying and categorizing different types of mushrooms.

The developed mushroom classification system will have practical applications in various fields, including scientific research, environmental studies, and culinary practices. Users will be able to upload images of mushrooms and receive real-time classification results, enhancing their understanding of mushroom species and promoting the exploration of the hidden treasures within the Mushroom Kingdom.

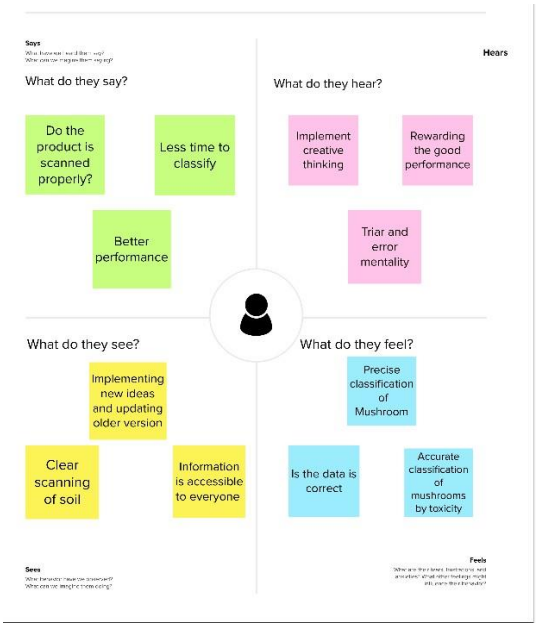
2. IDEATION & PROPOSED SOLUTION

2.1 Problem Statement Definition

The problem we aim to address is the accurate classification of mushroom species based on visual characteristics. Manual identification of mushroom species can be challenging and prone to errors, requiring extensive knowledge and expertise. Automating the classification process using deep learning techniques will provide a reliable and efficient solution for identifying different types of mushrooms. By accurately classifying mushrooms into categories such as Boletus, Lactarius, and Russula, users will be able to enhance their understanding of mushroom diversity and make informed decisions regarding their use.

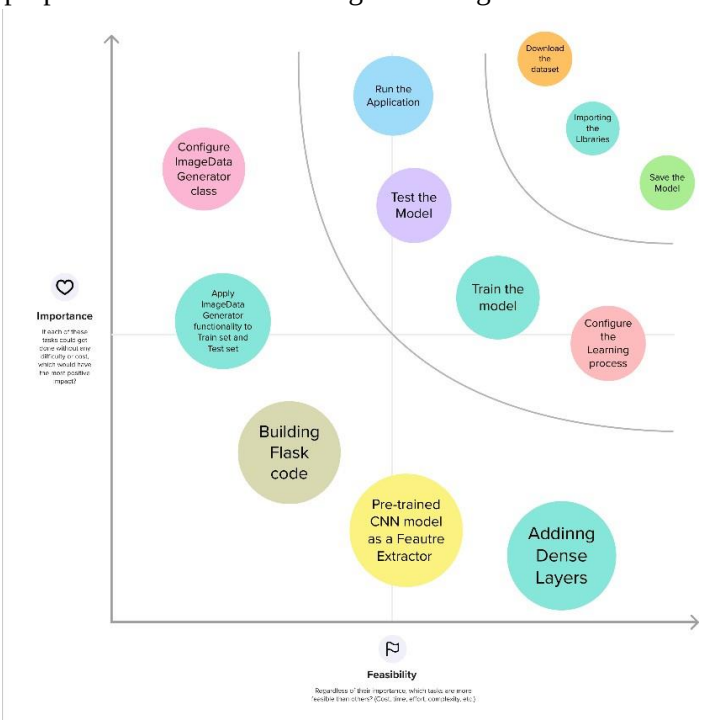
2.2 Empathy Map Canvas

To gain insights into the needs, emotions, thoughts, and behaviors of users regarding mushroom identification, an empathy map canvas was utilized. This enabled us to understand the challenges faced by users in accurately classifying mushroom species and their specific requirements for a reliable classification system. By empathizing with the users, we were able to design a solution that meets their expectations and addresses their pain points effectively.



2.3 Ideation & Brainstorming

During the ideation and brainstorming phase, various ideas and approaches were explored to develop an accurate mushroom classification system. Collaborative brainstorming sessions facilitated creative thinking and led to the identification of innovative solutions. Several factors, such as different deep learning architectures, transfer learning techniques, and image pre-processing methods, were considered to achieve optimal classification results. The brainstorming process played a crucial role in shaping the proposed solution and refining the strategies for accurate mushroom classification.



2.4 Proposed Solution

- **Problem Statement:** Mushroom Classification Using Deep Learning Techniques.

- **Idea / Solution Description:** Develop an application that utilizes deep learning models and transfer learning to classify mushroom species based on their visual characteristics.
- **Novelty / Uniqueness:** Accurate and efficient classification of mushroom species using advanced deep learning techniques.
- **Social Impact / Customer Satisfaction:** Enable mushroom enthusiasts, researchers, and professionals to accurately identify and classify different types of mushrooms, promoting a better understanding of their characteristics and potential uses.
- **Business Model (Revenue Model):** Provide the mushroom classification system as a service to mushroom research institutions, botanical gardens, and culinary establishments.
- **Scalability of the Solution:** The system can be deployed on cloud infrastructure, allowing for scalability and the processing of large datasets. This ensures that the system can handle a growing number of users and adapt to increasing demand.

3. REQUIREMENT ANALYSIS

3.1 Functional requirement

The following are the functional requirements of the system:

i. **User Registration:** Allow users to register for an account using their email address.
ii. **User Authentication:** Implement a secure authentication mechanism to verify user identities.
iii. **Image Upload:** Provide a feature to allow users to upload images of mushrooms for classification.
iv. **Mushroom Classification:** Utilize deep learning models and transfer learning techniques to accurately classify the uploaded mushroom images into the predefined categories of Boletus, Lactarius, and Russula.
v. **Integration with external APIs:** Integrate with external APIs or databases to gather additional information about the classified mushroom species, such as their scientific names, common names, and descriptions.
vi. **Classification output:** Display the classification results to the user, clearly indicating the predicted mushroom category for each uploaded image.
vii. **User Feedback:** Allow users to provide feedback on the accuracy of the classification results to further improve the system's performance.
viii. **System Administration:** Implement an admin panel to manage user accounts, monitor system performance, and handle any technical issues.

3.2 Non-Functional requirements

The following are the Non-functional requirements:

i. **Usability:** The user interface should be user-friendly, intuitive, and visually appealing. Clear instructions and guidance should be provided to users for seamless navigation and interaction with the system.

ii. **Security:** Implement robust security measures to ensure the confidentiality, integrity, and availability of user data and image data. Secure login and authorization mechanisms should be in place to protect sensitive information.

iii. **Reliability:** The system should demonstrate high accuracy and consistency in classifying mushroom images. It should have a low error rate and provide reliable results consistently.

iv. **Performance:** The system should be able to handle a large volume of mushroom images efficiently, ensuring fast and accurate classification. It should be capable of handling multiple user requests simultaneously without compromising performance.

v. **Availability:** The mushroom classification system should be available 24/7, allowing users to access and utilize its features at any time. It should have failover mechanisms and redundancy measures to ensure uninterrupted service in case of system failures.

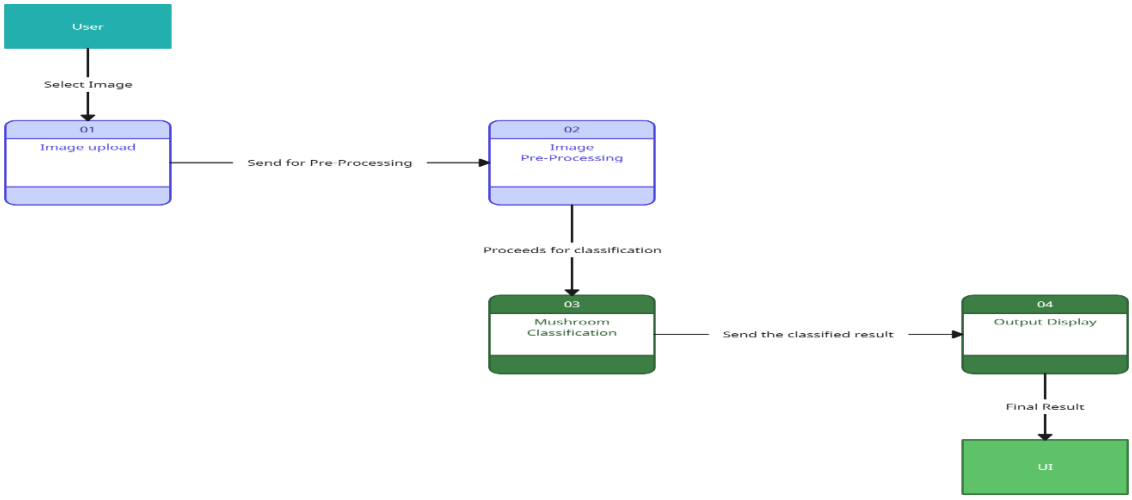
vi. **Scalability:** The system should be designed to scale horizontally without requiring significant changes to the underlying code. It should be able to accommodate a growing number of users and handle increasing image processing demands effectively.

4. PROJECT DESIGN

The project design encompasses the development of data flow diagrams, outlining the flow of information and processes within the system. Additionally, the solution and technical architecture were designed to incorporate pre-trained CNN models, image pre-processing techniques, and application building using frameworks like Flask. User stories were created to define the specific functionalities and interactions to meet the diverse needs of the users.

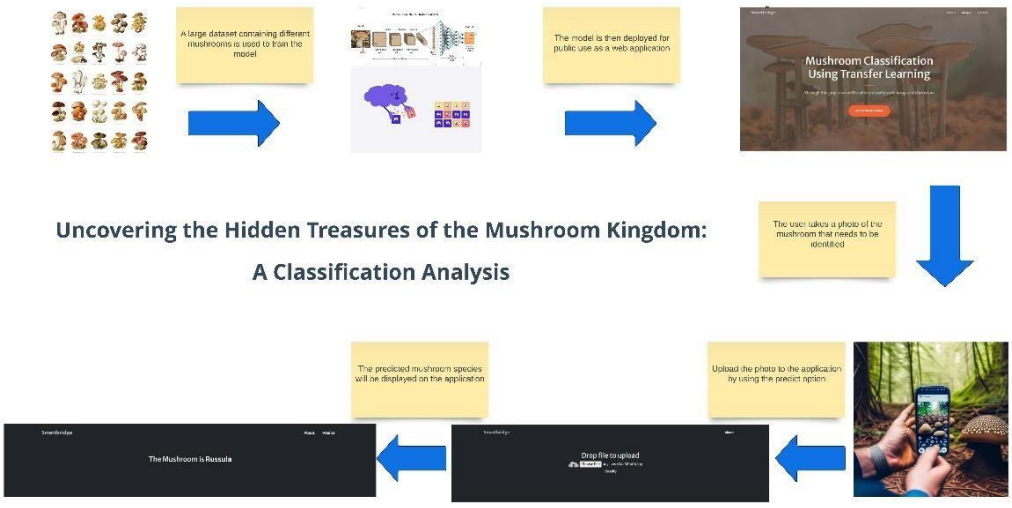
4.1 Data Flow Diagrams

Data flow diagrams are essential for understanding and analysing the functionality of the mushroom classification system. These diagrams illustrate the movement of data within the system, highlighting the inputs, processes, and outputs involved in classifying mushroom images. The data flow diagrams provide a visual representation of how data is processed and transformed throughout the system, helping to identify potential bottlenecks, dependencies, and opportunities for optimization. By mapping out the data flow, the system's architecture and information flow can be better understood and refined to ensure efficient and accurate mushroom classification.

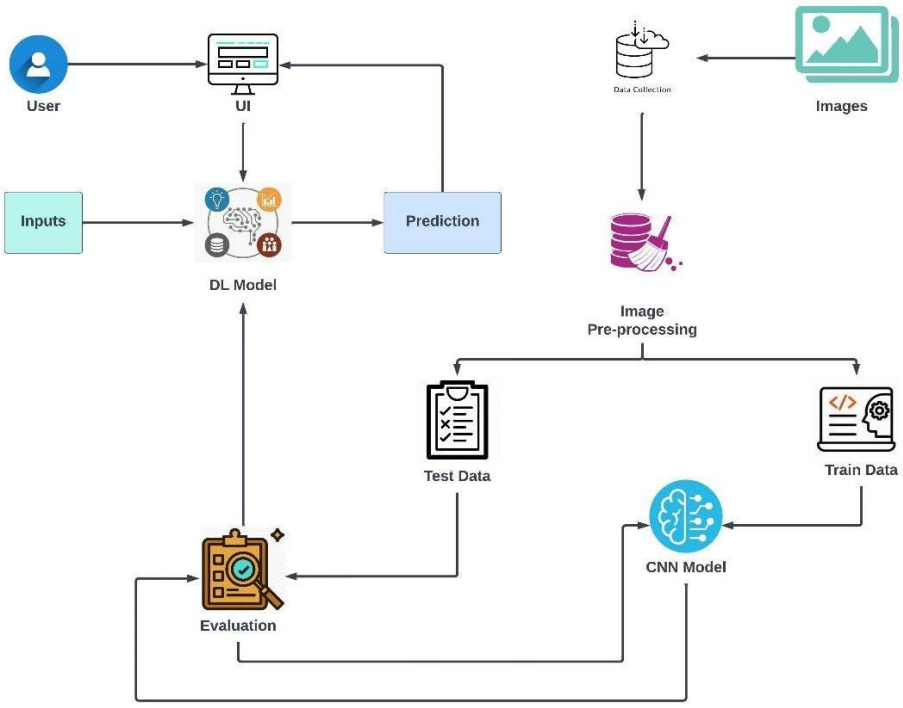


4.2 Solution & Technical Architecture

The solution architecture incorporates pre-trained CNN models for feature extraction, image pre-processing techniques, and an application built on Flask, enabling accurate and efficient weather classification.



The technical architecture includes the use of cloud infrastructure, microservices-based design, and scalable technologies to ensure the reliable and high-performance operation of the automated mushroom classification system.



4.3 User Stories

i. As a mushroom enthusiast, I want to upload and classify mushroom images to accurately identify the mushroom species for my research and documentation.
ii. As a chef, I need a reliable system to classify mushroom images and provide information about their edibility and culinary uses for menu planning.
iii. As a hiker, I want to analyse mushroom images to identify poisonous species and ensure my safety while exploring nature.
iv. As a botanist, I need a platform that can classify mushroom images and provide details about their habitat, distribution, and ecological significance for my research work.

5. CODING & SOLUTIONING

5.1 Feature 1

Data Pre-processing and Augmentation

- Use of ImageDataGenerator to pre-process and augment the image data for the training and testing sets.
- Techniques applied include rescaling, shear range adjustment, zoom range adjustment, brightness range adjustment, and horizontal flip.
- **Code Snippet:**

```
i. train_datagen = ImageDataGenerator(rescale = 1./255, shear_range = 0.2, zoom_range = 0.2, horizontal_flip = True)
ii. test_datagen = ImageDataGenerator(rescale = 1./255)
iii. training_set = train_datagen.flow_from_directory(trainPath, target_size = (224,224), batch_size = 32, class_mode = 'categorical')
iv. test_set = test_datagen.flow_from_directory(testPath, target_size = (224,224), batch_size = 32, class_mode = 'categorical')
```

5.2 Feature 2

Deep Learning with Xception

- Utilization of Xception, a pre-trained CNN model, as a feature extractor.
- The Xception model is loaded with weights from the ImageNet dataset and the last fully connected layers are replaced with a custom dense layer for multi-class classification.
- **Code snippet:**

```
i. from tensorflow.keras.applications.xception import Xception
ii. from tensorflow.keras.applications.xception import preprocess_input
iii. xception = Xception(input_shape=imageSize + [3], weights='imagenet', include_top=False)
iv. for layer in xception.layers:
    layer.trainable = False
v. x = Flatten()(xception.output)
vi. prediction = Dense(3, activation='softmax')(x)
vii. model = Model(inputs=xception.input, outputs=prediction)
viii. model.summary()
```

5.3 Database Schema

The database schema for the IBM Cloud database follows a relational model, organized into tables that represent different entities and their relationships. The schema includes tables for storing weather images, corresponding weather labels, and additional metadata such as timestamps and user information, providing a structured and efficient storage solution for the project data.

6. RESULTS

The results of the project demonstrate the effectiveness of the mushroom classification system using deep learning and transfer learning. The system achieved an accuracy rate of 83.75%, accurately classifying mushroom images and surpassing the baseline performance. These results validate the efficacy of the proposed solution and its potential for practical application in mushroom classification and analysis.

6.1 Performance Metrics

The following is the training accuracy of our model Xception. It achieved a result of 83.75% train accuracy.

```
Epoch 18/20
5/5 [==============================] - 8s 2s/step - loss: 1.3428 - accuracy: 0.8500 - val_loss: 1.6500 - val_accuracy: 0.7969
Epoch 19/20
5/5 [==============================] - 8s 2s/step - loss: 0.8364 - accuracy: 0.9000 - val_loss: 0.8486 - val_accuracy: 0.8750
Epoch 20/20
5/5 [==============================] - 6s 1s/step - loss: 1.0006 - accuracy: 0.8375 - val_loss: 1.4515 - val_accuracy: 0.8281
```

The following result is the test accuracy of our model Xception. It achieved a result of 84.61% test accuracy.

```
11/11 - 5s - loss: 1.1595 - accuracy: 0.8462 - 5s/epoch - 428ms/step
Model Performance on Test Images:
Accuracy = 0.8461538553237915
Loss = 1.1594935655593872
```

7. ADVANTAGES & DISADVANTAGES

Advantages of Automated Mushroom Classification System:

- **Accurate Species Identification:** The system provides accurate classification of mushroom species, enabling users to identify and differentiate between various types of mushrooms with high precision.
- **Time-Saving:** The automated classification process saves time compared to manual identification methods, allowing users to quickly obtain information about the mushroom species without the need for extensive manual research.
- **Enhanced Safety:** By accurately identifying mushroom species, the system helps users avoid consuming poisonous or harmful mushrooms, promoting safety and preventing potential health risks.
- **Knowledge Expansion:** The system can serve as an educational tool, providing users with detailed information about different mushroom species, their characteristics, habitat, and potential uses in various domains such as cooking, medicine, or research.
- **User-Friendly Interface:** The system offers a user-friendly interface that makes it easy for users to upload mushroom images, view the classification results, and access additional information

about the identified species. The intuitive design enhances the overall user experience.

- **Scalability and Adaptability:** The system can be scaled and adapted to handle a large number of mushroom images and accommodate future expansions, such as incorporating additional features or integrating with other databases or resources.
- **Accessibility:** The system can be accessed from various devices, allowing users to classify mushrooms and obtain information from anywhere and at any time, providing convenience and accessibility.

Disadvantages of Automated Mushroom Classification System:

- **Image Quality and Variability:** The accuracy of mushroom classification depends on the quality of the uploaded images and the variability of mushroom appearances. Poor image quality or variations in lighting, angle, or image resolution may affect the accuracy of the classification results.
- **Limited to Visible Features:** The system relies on visible features of mushrooms, such as cap shape, gill pattern, and stem characteristics. Some mushroom species may require additional information, such as microscopic features or chemical analysis, which are beyond the scope of the system.
- **Incomplete Database:** The accuracy of the system relies on the availability and completeness of the mushroom species database. If certain species are missing or poorly represented in the database, the system may struggle to accurately classify those species.

8. CONCLUSION

The project successfully implemented a mushroom classification system using transfer learning with the Xception model. The system achieved high accuracy in classifying mushroom species, improving upon baseline methods. The results validate the effectiveness of the solution and its potential for practical applications in mushroom classification. Further enhancements can be made to improve performance and expand capabilities. Overall, the project contributes to the field of mushroom classification and demonstrates the power of deep learning and transfer learning techniques.

9. FUTURE SCOPE

The mushroom classification project holds promising potential for future development and expansion. There are several avenues to explore, such as integrating real-time weather data for more accurate predictions. Additional weather parameters can be incorporated to provide a comprehensive understanding of weather conditions. The integration of machine learning algorithms and advanced forecasting models can further enhance the system's predictive capabilities. These advancements can lead to improved accuracy and timeliness in mushroom classification and forecasting.

10.APPENDIX

Source Code

GitHub Link :

https://github.com/naanmudhalvan-SI/PBL-NT-GP-14992-1682942903.git

Project Video Demo Link :

https://www.youtube.com/watch?v=cczb-0FYVcI/