# Project Report

# INTRODUCTION

## 1.1 Project Overview

Uncovering The Hidden Treasures of "The Mushroom Kingdom", mushrooms are a type of fungi that grow in a variety of habitats, from forests to fields to decomposing logs. They come in many different shapes, sizes, and colors, and are used for food, medicine, and other purposes. A mushroom is a fruitful body of fungus which is usually produced above the ground on soil or other nutrients. We only concentrate on mushrooms and do not consider fungus species. Our purpose is the optical recognition of species which have cap, gills underside of cap and astern and typically the full body is visible on the image.

In this project we are classifying various types of Mushrooms that are found on various regions of our planet. These Mushrooms are majorly classified into 3 categories namely Boletus, Lactarius & Russula. Deep-learning (DL) methods in artificial intelligence (AI) play a dominant role as high-performance classifiers in the detection of the Mushrooms using images. Transfer learning has become one of the most common techniques that has achieved better performance in many areas, especially in image analysis and classification. We used Transfer Learning techniques like Inception V3,Resnet50V2, Xception that are more widely used as a transfer learning method in image analysis and they are highly effective.

By systematically analyzing passages, researchers can discover patterns and relationships, and can uncover hidden secrets in the scale kingdom This research can include exploring treasures, such as rarity, efficiency, or impact of play. It can also explore any connection between a particular treasure and the game’s mechanics, level design, or narrative.

The findings from this segmentation study can help to provide a deeper understanding of Mushroom Kingdom design, improve player gameplay experiences, or provide insights for future game development. The main goal of the project is to develop a mushroom identification system that can accurately identify and classify three major mushroom species: Boletus, Lactarius, and Russula This taxonomic analysis can contribute to a better understanding of mushroom taxonomy

## 1.2 Purpose

The purpose of the project "Uncovering The Hidden Treasures of the Mushroom Kingdom: A Classification Analysis" is to develop an optical recognition system that can accurately classify different types of mushrooms found in various regions. The project aims to classify mushrooms into three major categories: Boletus, Lactarius, and Russula.

By utilizing deep-learning techniques and transfer learning, the project seeks to train models to accurately identify and categorize these mushroom species based on their visual characteristics.

High-Performance Classification: The project focuses on achieving high-performance classification by leveraging pre-trained models that have been trained on large image datasets. By fine-tuning these models and training them on specific mushroom data, the goal is to develop a robust and accurate optical recognition system.

Application in Image Analysis and Classification: The project utilizes deep-learning methods, such as Inception V3, ResNet50V2, and Xception, which are widely used and highly effective in image analysis and classification tasks. By applying these techniques to mushroom classification, the project aims to contribute to the advancement of image analysis and classification in the field of mycology.

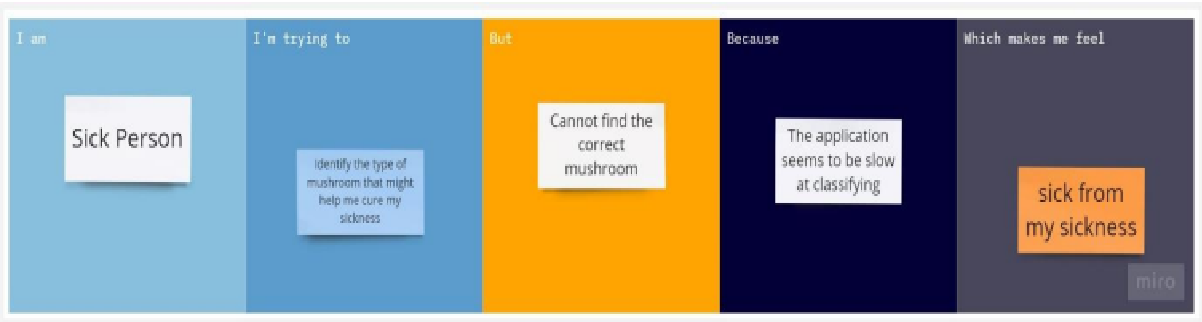
Contribution to Mushroom Research and Applications: The accurate classification of mushroom species can have various implications. It can assist in ecological studies by providing insights into species distribution and diversity. It can also aid in conservation efforts by helping to identify and protect endangered or rare mushroom species. Additionally, the classification system can be used by mushroom enthusiasts, mycologists, or even culinary explorers to identify mushrooms for scientific, educational, or culinary purposes. Overall, the purpose of the project is to develop an efficient and accurate optical recognition system for mushroom classification, which can contribute to the advancement of mycology research, conservation efforts, and practical applications involving mushrooms.

## IDEATION & PROPOSED SOLUTION

# 2.1 Problem Statement Definition

| | |
|---|---|
| Date | 02 May 2023 |
| Team ID | NM2023TMID15724 |
| Project Name | Uncovering the Hidden Treasures of the Mushroom Kingdom: A Classification Analysis |
| Maximum Marks | 2 Marks |

**Customer Problem Statement:**




# 2.2 Empathy Map Canvas

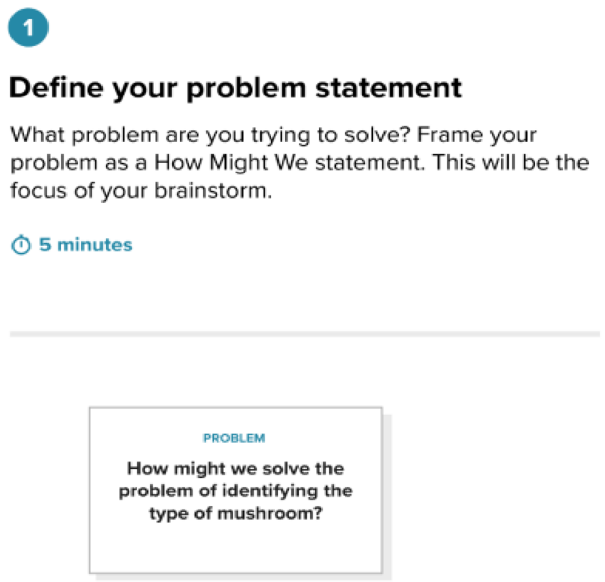
| | |
|---|---|
| Date | 02 May 2023 |
| Team ID | NM2023TMID15724 |
| Project Name | Uncovering the Hidden Treasures of the Mushroom Kingdom: A Classification Analysis |
| Maximum Marks | 4 Marks |

**Empathy Map for the Project:**

# 2.3 Ideation & Brainstorming

| | |
|---|---|
| Date | 03 May 2023 |
| Team ID | NM2023TMID15724 |
| Project Name | Uncovering the Hidden Treasures of the Mushroom Kingdom: A Classification Analysis |
| Maximum Marks | 4 Marks |

**Brainstorm & Idea Prioritization:**




## Step-2: Brainstorm, Idea Listing and Grouping

2

### Brainstorm

Write down any ideas that come to mind
that address your problem statement.

10 minutes

TIP

You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

**Gowtham M**

- Create a web Application
- Use deep learning

**Daanish**

- Using Flask Framework
- Each user with his own uploaded pics

**Nandhakumar**

- Object detection using CNN
- Show all the information about the Mushroom

**Mohammed Harshath**

- Also make a mobile app
- Simple UI

**Abhijith**

- User Feedback
- Faster computation

3

### Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes

TIP

Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.

**Application**

- Flask Framework
- Web Application
- Simple UI

**CNN**

- convolutional neural network
- tensorflow
- Python

**Storage**

- IBM Cloud

## Step-3: Idea Prioritization

4

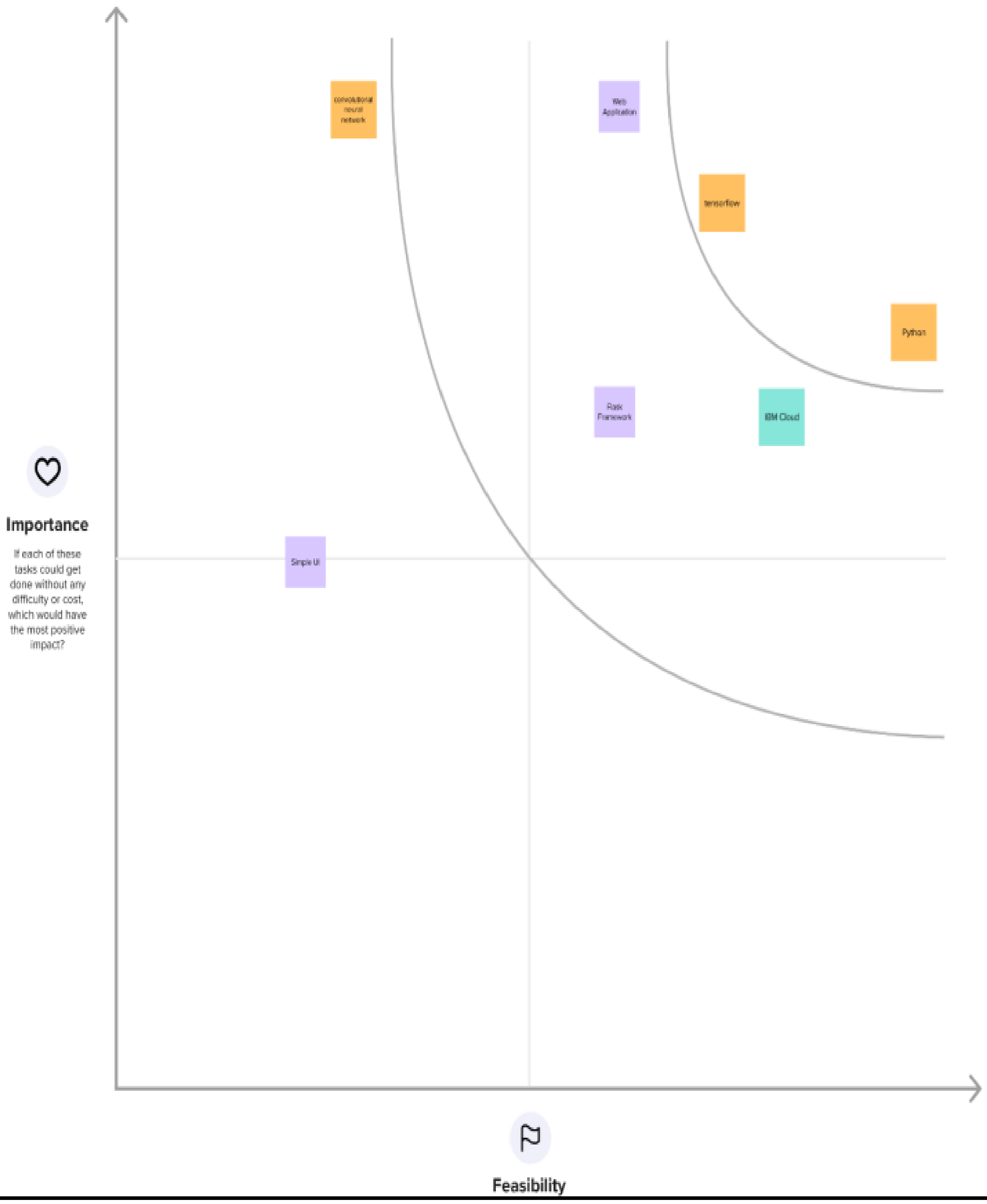
### Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes

**TIP**

Participants can use their cursors to point at where sticky notes should go on the grid. The facilitator can confirm the spot by using the laser pointer holding the **H key** on the keyboard.



## 2.4 Proposed Solution

| | |
|---|---|
| Date | 16 May 2023 |
| Team ID | NM2023TMID15724 |
| Project Name | Uncovering the Hidden Treasures of the Mushroom Kingdom: A Classification Analysis |

**Proposed Solution :**

| S.No. | Parameter | Description |
|---|---|---|
| 1. | Problem Statement (Problem to be solved) | In this project we are classifying various types of Mushrooms that are found on various regions of our planet. |
| 2. | Idea / Solution description | We used Deep Learning to train the model that will classify the image. |
| 3. | Novelty / Uniqueness | We are using the Xception pretrained deep learning model. |
| 4. | Social Impact / Customer Satisfaction | Can help the user to identify the type of mushroom that from an image uploaded into the application. |
| 5. | Business Model (Revenue Model) | Can be helpful for people living near forests areas and depend on it for food |
| 6. | Scalability of the Solution | Can be Scaled into an application that can also identify specific name of the mushroom as it can only calculate the type of class it belongs to. |

## REQUIREMENT ANALYSIS

# 3.1 Functional requirement

Following are the functional requirements of the proposed solution.

| FR No. | Functional Requirement (Epic) | Sub Requirement (Story / Sub-Task) |
|---|---|---|
| FR-1 | Image Input | Obtaining the image through HTML<br>Saving the Image Locally |
| FR-2 | Classified Output | Classify the image using the Model<br>Send the output back to HTML |

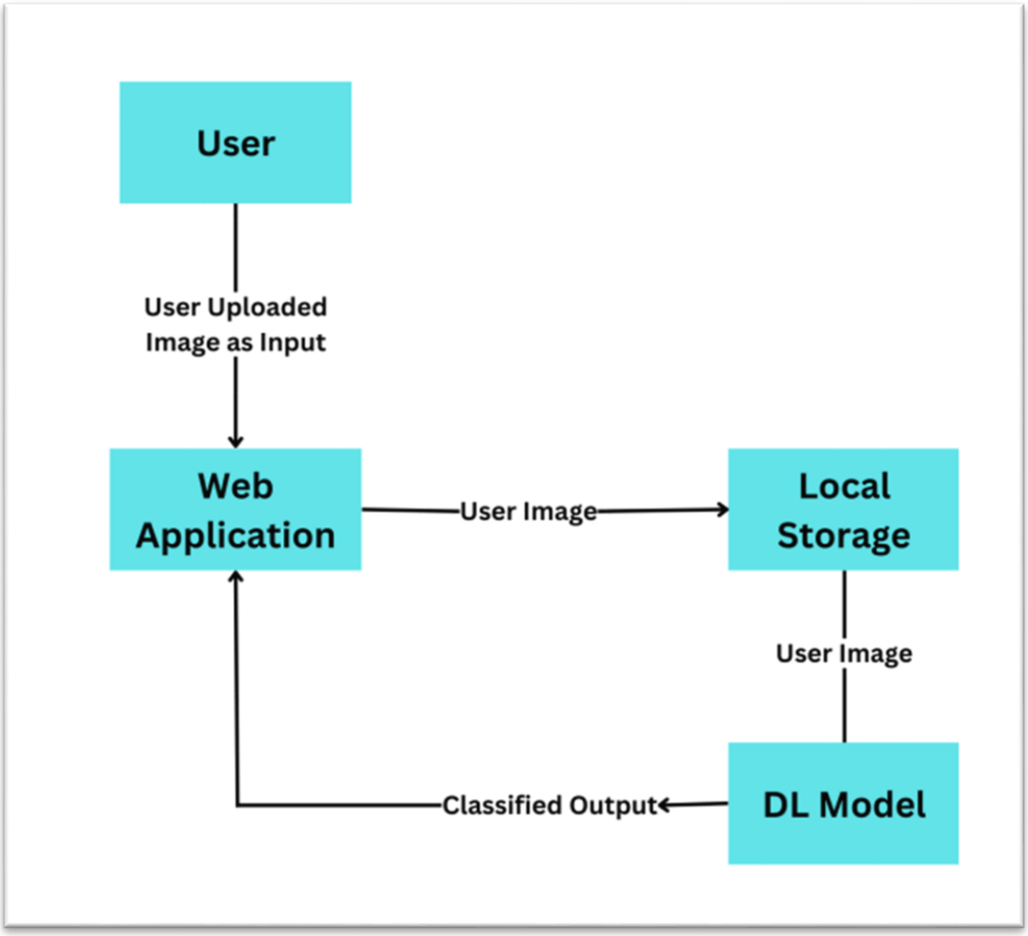
# 3.2 Non-Functional requirements

Following are the non-functional requirements of the proposed solution.

| FR No. | Non-Functional Requirement | Description |
|---|---|---|
| NFR-1 | **Usability** | Can be used by anyone who wants to know the type of mushroom |
| NFR-2 | **Security** | No security required |
| NFR-3 | **Reliability** | Uses the Xception , So very much accurate |
| NFR-4 | **Performance** | Fast Classification |
| NFR-5 | **Availability** | For everyone |
| NFR-6 | **Scalability** | Can be made into a mobile app as well |

# PROJECT DESIGN

## 4.1 Data Flow Diagrams




## 4.2 Solution & Technical Architecture

**Table-1 : Components & Technologies:**

| S.No | Component | Description | Technology |
|---|---|---|---|
| 1. | User Interface | Web UI | HTML, CSS, JavaScript |
| 2. | Application Logic-1 | Flask for building the web app | Python |
| 3. | Application Logic-2 | Training the model | Jupyter |
| 4. | Application Logic-3 | Training the model on cloud | IBM Watson Machine Learning |
| 5. | Machine Learning Model | Classify the type of Mushroom | Image Classification Model |
| 6. | Infrastructure (Server / Cloud) | Application Deployment on Local System | Local |

**Table-2: Application Characteristics:**

| S.No | Characteristics | Description | Technology |
|---|---|---|---|
| 1. | Open-Source Frameworks | Python, Jupyter, Tensorflow, Flask | Technology of Opensource framework |
| 2. | Security Implementations | No Security | Null |
| 3. | Scalable Architecture | Can implement database | IBM database |
| 4. | Availability | Web Application | Flask |
| 5. | Performance | Stores the images locally | Local storage |

# 4.3 User Stories

| User Type | Functional Requirement (Epic) | User Story Number | User Story / Task | Acceptance criteria | Priority | Team Member |
|---|---|---|---|---|---|---|
| Customer (Web User) | Uploading Image | USN-1 | As a user, I can upload a mushroom image | I can get the type of class | High | Gowtham |
| | Home Page | USN-2 | As a user, I can know about mushrooms | I can see the information available at the home page | High | Abhijith |
| | Uploading Image | USN-3 | As a user, I can upload several mushroom images one after the other | I can upload multiple mushroom images one after the other | High | Harshath |
| | Login | USN-4 | As a user, I don't require to login | Application does not require login | Medium | Nandhakumar |
| | Home Page | USN-5 | As a user, I can view the video about mushrooms | Home page has a video link about mushrooms | Low | Daanish |

# CODING & SOLUTIONING

## 5.1 Feature 1

### Training the model

We have used the Xception model to train our Mushroom dataset which contains three classes Boletus, Lactarius, Russula. After the model is trained we save the model in H5-file(Hierarchical data format).

## 5.2 Feature 2

### Building the web application

We use Flask framework to build our application. We build the user interface with HTML , CSS and JavaScript and integrated the web frame using Flask . When the user gives an input image the image is preprocessed and classified using the Xception model that is trained and loaded into the flask webframe .

# RESULTS

## 6.1 Performance Metrics

| S.No. | Parameter | Values | Screenshot |
|---|---|---|---|
| 1. | Model Summary | Layers in Xception Model with an extra flatten and dense layer | flatten_1 (Flatten) (None, 100352)<br>dense (Dense) (None, 3)<br>==============================================<br>Total params: 21,162,539<br>Trainable params: 301,059<br>Non-trainable params: 20,861,480 |
| 2. | Accuracy | Training Accuracy – 0.8750<br><br>Validation Accuracy – 0.8906 | Epoch 20/20<br>5/5 [==============================] - 22s 4s/step<br>loss: 0.7332 - accuracy: 0.8750 - val_loss: 0.7901<br>val_accuracy: 0.8906 |

# ADVANTAGES & DISADVANTAGES

## ADVANTAGES

- **Diversity:**

  The work focuses on different visual scales, which may have various practical applications. These include edible poisonous mushrooms, assisting in environmental research and assisting in mushroom production.

- **Efficient classification:**

  Using deep learning techniques such as transfer learning with models such as Inception V3, ResNet50V2, and Xception, the project can achieve high performance classification. Transfer learning applies pre-trained models to large image datasets, resulting in fast and accurate classification of scales

- **Comprehensiveness:**

  The project aims to classify mushrooms from different regions of the planet, suggesting a comprehensive approach to the study of mushroom species. This may contribute to a better understanding of mushroom distribution and species worldwide.

- **Automation and Scalability:**

  Using deep learning techniques, the project enables automation of scale allocation based on image analysis. This scalability allows for the efficient handling of large numbers of scale models, and can facilitate the development of complete inventories of scales.

## DISADVANTAGES

- **Limitation to seen functions:**

  The project focuses on optical recognition, specifically on the cap, gills underside, and stem of mushrooms. While these are important features for identification, certain species may have additional distinguishing characteristics that cannot be captured solely through images. This limitation may result in misclassification or incomplete identification of some mushroom species.

- **Dataset barriers:**

  The accuracy and performance of deep learning models heavily rely on the quality and diversity of the training dataset. If the dataset used for training is limited in terms of the number of mushroom species or the quantity of representative images, the classification accuracy may be compromised. Ensuring a large and diverse dataset can be challenging, especially for rare or lesser-known mushroom species.

- **Interpretability:**

  Deep learning models, including transfer learning techniques, are often referred to as "black boxes" because their decision-making processes can be difficult to interpret. Understanding the rationale behind the classification results may be challenging, which could limit the ability to validate or explain the model's predictions

# CONCLUSION

In conclusion, the project focuses on the optical recognition of mushroom species using deep learning techniques, specifically employing transfer learning with models like Inception V3, ResNet50V2, and Xception. This approach offers several advantages, including efficient classification, a wide range of applications, broad coverage of mushroom diversity, and the potential for automation and scalability. However, there are also limitations to consider, such as the reliance on visible features, potential dataset limitations, challenges in interpretability, and dependency on image quality. Despite these limitations, the project holds promise for advancing the classification and understanding of various mushroom species found across different regions of the planet.

## FUTURE SCOPE

**1. Integration of additional features:**

Expanding the scope of the project to consider other important features of mushrooms, beyond the cap, gills underside, and stem, can improve the accuracy of classification. For example, incorporating texture analysis, spore color, or other macroscopic and microscopic characteristics can provide a more comprehensive understanding of mushroom species.

2. **Real-time mushroom identification**:

Expanding the project to develop a mobile or web application that allows users to identify mushrooms in real-time using their smartphones or cameras can be a valuable tool for mushroom enthusiasts, hikers, or foragers. The application can leverage the trained deep learning models and provide instant species recognition along with relevant information and safety warnings.

3. **Collaboration and crowd-sourcing:**

Encouraging collaboration and crowd-sourcing efforts can help in expanding the dataset and improving the accuracy of the classification models. Collecting images from citizen scientists, mushroom foragers, and researchers worldwide can contribute to a more diverse and representative dataset, enhancing the project's capabilities.

# APPENDIX

## Mushroom_Classification_Using_Xception_IBM.ipynb

```
import os, types
import pandas as pd
from botocore.client import Config
import ibm_boto3
def __iter__(self): return 0


# @hidden_cell
# The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.
# You might want to remove those credentials before you share the notebook.
cos_client = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='3l4zywi_QpOVMR0leMN7--DPNZQFnfg5Ie2GPlYAjmJZ',
    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token", config=Config(signature_version='oauth'),endpoint_url='https://s3.private.us-south.cloud-object-storage.appdomain.cloud')


bucket = 'mushroomclassification-donotdelete-pr-zu4bsjlmtup6r8'
object_key = 'Dataset.zip'


streaming_body_2 = cos_client.get_object(Bucket=bucket, Key=object_key)['Body']
import zipfile
from io import BytesIO
zip_ref = zipfile.ZipFile(BytesIO(streaming_body_2.read()), 'r')
zip_ref.extractall('')
```

```
zip_ref.close()

imageSize = [224, 224]

trainPath = r'Dataset/train'

testPath = r'Dataset/test'

from tensorflow.keras.layers import Dense, Flatten, Input

from tensorflow.keras.models import Model

from tensorflow.keras.preprocessing import image

from tensorflow.keras.preprocessing.image import ImageDataGenerator,
load_img

from tensorflow.keras.applications.xception import Xception,
preprocess_input

from glob import glob

import numpy as np

import matplotlib.pyplot as plt

train_datagen = ImageDataGenerator(rescale = 1./255, shear_range = .2,
zoom_range = .2, horizontal_flip = True)

test_datagen = ImageDataGenerator(rescale = 1./255)

train_set = train_datagen.flow_from_directory(trainPath, target_size = (224,
224), batch_size = 32, class_mode = 'categorical')

test_set = test_datagen.flow_from_directory(testPath, target_size = (224,
224),batch_size = 32, class_mode = 'categorical')

xception = Xception(input_shape = imageSize + [3] , weights = 'imagenet',
include_top=False)

for layer in xception.layers:

   layer.trainable = False

x = Flatten()(xception.output)

prediction = Dense(3, activation= 'softmax')(x)

model = Model(inputs = xception.inputs, outputs = prediction)
```

```
model.summary()

model.compile(loss = 'categorical_crossentropy', optimizer = 'adam', metrics = ['accuracy'])

#fit the model

r = model.fit(train_set, validation_data = test_set, epochs = 20, steps_per_epoch = len(train_set)//5, validation_steps = len(test_set)//5 )

model.save('mushroom.h5')

train_set.class_indices

img = image.load_img(r"Dataset/test/Lactarius/011_LoMY4qu_wxo.jpg", target_size = imageSize)

x = image.img_to_array(img)

x.shape

import numpy as np

x = np.expand_dims(x, axis = 0)

img_data = preprocess_input(x)

img_data.shape

output = np.argmax(model.predict(img_data), axis = 1)

output
```

## IBM Deployment

```
pip install watson-machine-learning-client

pip install ibm-watson-machine-learning

from ibm_watson_machine_learning import APIClient

wml_credentials = {
   "url":"https://us-south.ml.cloud.ibm.com",
   "apikey":"qOuHGMi-Vc7V7otwRjKdAuJrstuswSIio8N3LssOOed_"
}
```

```python
client = APIClient(wml_credentials)

def uid_space_name(client, mushroom_deploy):
   space = client.spaces.get_details()
   return (next(item for item in space['resources'] if item['entity']['name'] == mushroom_deploy)['metadata']['id'])

space_uid = uid_space_name(client, 'mushroom_deploy')

client.set.default_space(space_uid)

client.software_specifications.list(100)

software_space_uid = client.software_specifications.get_uid_by_name('tensorflow_rt22.2-py3.10')

!tar zcvf mushroom.tgz mushroom.h5

software_space_uid

import tensorflow as tf

tf.__version__

#Storing the MODEL

model_details = client.repository.store_model(model = 'mushroom.tgz',
                                              meta_props = {
client.repository.ModelMetaNames.NAME:"Mushroom_CNN_Model",
client.repository.ModelMetaNames.TYPE:'tensorflow_2.9',
client.repository.ModelMetaNames.SOFTWARE_SPEC_UID:software_space_uid})

model_id = client.repository.get_model_id(model_details)

model_id

#Downloading the model through the client

client.repository.download('4e717c56-db84-4666-9d14-ea8b3a38f686', 'mush.tgz')
```

## Flask – app.py

```
import numpy as np
import os
import sys
from flask import Flask, request, render_template
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
from tensorflow.keras.applications.xception import preprocess_input
from PIL import *
model = load_model('.\Flask\mushroom.h5')
app = Flask(__name__)
@app.route('/')
def index():
   return render_template('index.html')
@app.route('/home')
def home():
   return render_template('index.html')
@app.route('/input')
def input():
   return render_template('predict.html')
@app.route('/predict', methods = ["GET", "POST"])
def res():
   if request.method=="POST":
      f = request.files['images']
      basepath = os.path.dirname(__file__)
      filepath = os.path.join(basepath, 'uploads', f.filename)
```

```
        filepath2 = os.path.join(basepath, 'static','uploads', f.filename)
        f.save(filepath)
        img1 = image.load_img(filepath)
        img = image.load_img(filepath, target_size = (224, 224, 3))
        img1.save(filepath2)
        x = image.img_to_array(img)
        x = np.expand_dims(x, axis = 0)
        img_data = preprocess_input(x)
        prediction = np.argmax(model.predict(img_data), axis = 1)
        index = ['Boletus', 'Lactarius', 'Russula']
        result = str(index[prediction[0]])
        print(result)
        return render_template('output.html', prediction = result, fname =
f.filename)
if __name__=="__main__":
    app.run(debug=True)
```