**PROJECT REPORT**

**PROJECT TITLE:**

**Pollen’s Profiling: Automated Classification Of Pollen Grains**

import tensorflow as tf

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense

from tensorflow.keras.preprocessing.image import ImageDataGenerator

import matplotlib.pyplot as plt

# Step 1: Data Loading and Preprocessing

train\_path = 'dataset/train'

test\_path = 'dataset/test'

train\_datagen = ImageDataGenerator(rescale=1./255)

test\_datagen = ImageDataGenerator(rescale=1./255)

train\_data = train\_datagen.flow\_from\_directory(

train\_path,

target\_size=(100, 100),

batch\_size=32,

class\_mode='categorical'

)

test\_data = test\_datagen.flow\_from\_directory(

test\_path,

target\_size=(100, 100),

batch\_size=32,

class\_mode='categorical'

)

# Step 2: CNN Model

model = Sequential([

Conv2D(32, (3, 3), activation='relu', input\_shape=(100, 100, 3)),

MaxPooling2D(2, 2),

Conv2D(64, (3, 3), activation='relu'),

MaxPooling2D(2, 2),

Flatten(),

Dense(128, activation='relu'),

Dense(train\_data.num\_classes, activation='softmax')

])

model.compile(optimizer='adam', loss='categorical\_crossentropy', metrics=['accuracy'])

# Step 3: Model Training

history = model.fit(train\_data, epochs=10, validation\_data=test\_data)

# Step 4: Model Evaluation

loss, acc = model.evaluate(test\_data)

print(f"Test Accuracy: {acc\*100:.2f}%")

# Step 5: Plot Accuracy and Loss

plt.plot(history.history['accuracy'], label='train\_acc')

plt.plot(history.history['val\_accuracy'], label='val\_acc')

plt.title('Model Accuracy')

plt.xlabel('Epochs')

plt.ylabel('Accuracy')

plt.legend()

plt.show()

**OUTPUT:**

Found 500 images belonging to 3 classes.

Found 150 images belonging to 3 classes.

Epoch 1/10

...

Epoch 10/10

Test Accuracy: 91.33%

**FINAL LINK:**

https://github.com/charishma/PollenGrainClassification\_Final\_IO\_project