

**Data collection and preparation**

**1.Data Collection:**

Dataset to use: RiceImage Dataset

Search for “Rice Image Dataset” on Kaggle for uploaded versions or enhanced features

**2.Data Preparation:**

**Code for (GrainPalette – Rice Type Classification Network**)

import os

import numpy as np

import matplotlib.pyplot as plt

from sklearn.metrics import classification\_report, accuracy\_score

from tensorflow.keras.models import Sequential

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

from tensorflow.keras.preprocessing.image import ImageDataGenerator

from tensorflow.keras.optimizers import Adam

# Data Preparation

img\_height, img\_width = 150, 150

batch\_size = 32

data\_dir = "/path\_to\_dataset/Rice\_Image\_Dataset"

datagen = ImageDataGenerator(rescale=1./255, validation\_split=0.2)

train\_generator = datagen.flow\_from\_directory(

data\_dir,

target\_size=(img\_height, img\_width),

batch\_size=batch\_size,

class\_mode='categorical', subset='training'

)

val\_generator = datagen.flow\_from\_directorydata\_dir,

target\_size=(img\_height, img\_width),

batch\_size=batch\_size,

class\_mode='categorical',

subset='validation'

)

# CNN Model

model = Sequential([

Conv2D(32, (3, 3), activation='relu', input\_shape=(img\_height, img\_width, 3)),

MaxPooling2D(pool\_size=(2, 2)),

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

MaxPooling2D(pool\_size=(2, 2)),

Flatten(),

Dense(128, activation='relu'),

Dropout(0.5),

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

])

model.compile(optimizer=Adam(learning\_rate=0.0001), loss='categorical\_crossentropy', metrics=['accuracy'])

# Training

model.fit(train\_generator, epochs=10, validation\_data=val\_generator)

# Evaluation

y\_pred = model.predict(val\_generator)

y\_pred\_classes = np.argmax(y\_pred, axis=1)

y\_true = val\_generator.classes

print("Accuracy:", accuracy\_score(y\_true, y\_pred\_classes))

print("Classification Report:\n", classification\_report(y\_true, y\_pred\_classes))

# Save the Model

model.save("cnn\_rice\_classifier.h5")