# Role: AI/ML Engineer

# Module: Food Fingerprinting (AI-based Authenticity Verification)

import cv2

import numpy as np

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

# Step 1: Load and Preprocess Data

def load\_data(directory):

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

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

directory,

target\_size=(128, 128),

batch\_size=32,

class\_mode='binary',

subset='training'

)

validation\_data = datagen.flow\_from\_directory(

directory,

target\_size=(128, 128),

batch\_size=32,

class\_mode='binary',

subset='validation'

)

return train\_data, validation\_data

# Step 2: Define AI Model

def build\_model():

model = Sequential([

Conv2D(32, (3,3), activation='relu', input\_shape=(128,128,3)),

MaxPooling2D(2,2),

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

MaxPooling2D(2,2),

Flatten(),

Dense(128, activation='relu'),

Dense(1, activation='sigmoid')

])

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

return model

# Step 3: Train the Model

def train\_model(model, train\_data, validation\_data, epochs=10):

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

model.save("food\_fingerprint\_model.h5")

# Step 4: Predict on New Image

def predict\_fingerprint(model\_path, image\_path):

model = tf.keras.models.load\_model(model\_path)

image = cv2.imread(image\_path)

image = cv2.resize(image, (128, 128))

image = np.expand\_dims(image, axis=0) / 255.0

prediction = model.predict(image)

return "Authentic" if prediction > 0.5 else "Fraudulent"

# Running the Pipeline

if \_\_name\_\_ == "\_\_main\_\_":

train\_data, validation\_data = load\_data("dataset/food\_images")

model = build\_model()

train\_model(model, train\_data, validation\_data)

result = predict\_fingerprint("food\_fingerprint\_model.h5", "dataset/test\_image.jpg")

print(f"Food Authentication Result: {result}")