```
# =========================================================
# WEEK 10 — ADVANCED DEEP LEARNING (CNN)
# Student Performance Prediction
# Transfer Learning using MobileNetV2
# =========================================================

import tensorflow as tf
from tensorflow.keras.applications import MobileNetV2
from tensorflow.keras.layers import Dense, GlobalAveragePooling2D, Dropout
from tensorflow.keras.models import Model
from tensorflow.keras.preprocessing.image import ImageDataGenerator

# ------------------------------------
# Step 1: Data Preparation
# ------------------------------------
# Student images are divided into classes:
# Fail, Pass, Good, Excellent

img_size = (224, 224)
batch_size = 32

datagen = ImageDataGenerator(
   rescale=1./255,
   validation_split=0.2
)

train_data = datagen.flow_from_directory(
   "dataset/student_performance",
   target_size=img_size,
   batch_size=batch_size,
   class_mode="categorical",
   subset="training"
)

val_data = datagen.flow_from_directory(
   "dataset/student_performance",
   target_size=img_size,
   batch_size=batch_size,
   class_mode="categorical",
   subset="validation"
)

# ------------------------------------
# Step 2: Pre-trained CNN Model
# ---------------------------------
base_model = MobileNetV2(
    weights="imagenet",
    include_top=False,
    input_shape=(224, 224, 3)
)

# Freeze base layers
base_model.trainable = False

# ---------------------------------
# Step 3: Custom Classification Head
# ---------------------------------
x = base_model.output
x = GlobalAveragePooling2D()(x)
x = Dropout(0.3)(x)
x = Dense(256, activation="relu")(x)
outputs = Dense(4, activation="softmax")(x)

model = Model(inputs=base_model.input, outputs=outputs)

# ---------------------------------
# Step 4: Compile Model
# ---------------------------------
model.compile(
    optimizer="adam",
    loss="categorical_crossentropy",
    metrics=["accuracy"]
)

# ---------------------------------
# Step 5: Model Training
# ---------------------------------
model.fit(
    train_data,
    epochs=10,
    validation_data=val_data
)

# ---------------------------------
# Step 6: Model Summary
# ---------------------------------
model.summary()
```