import numpy as np

import tensorflow as tf

from tensorflow.keras.applications import InceptionV3

from tensorflow.keras.models import Model

from tensorflow.keras.preprocessing import image

from tensorflow.keras.preprocessing.sequence import pad\_sequences

from tensorflow.keras.preprocessing.text import Tokenizer

from tensorflow.keras.layers import Input, Dense, LSTM, Embedding, add

import pickle

# Load pre-trained CNN (InceptionV3) for feature extraction

def extract\_features(img\_path):

model = InceptionV3(weights='imagenet')

model = Model(model.input, model.layers[-2].output)

img = image.load\_img(img\_path, target\_size=(299, 299))

img = image.img\_to\_array(img)

img = np.expand\_dims(img, axis=0)

img = tf.keras.applications.inception\_v3.preprocess\_input(img)

feature = model.predict(img)

return feature

# Dummy tokenizer and model for captioning (Normally train these on a dataset like Flickr8k)

tokenizer = Tokenizer()

tokenizer.word\_index = {'startseq': 1, 'a': 2, 'dog': 3, 'is': 4, 'running': 5, 'in': 6, 'grass': 7, 'endseq': 8}

vocab\_size = len(tokenizer.word\_index) + 1

max\_length = 5

# Define the image captioning model

def define\_caption\_model():

inputs1 = Input(shape=(2048,)) # Image feature vector

fe1 = Dense(256, activation='relu')(inputs1)

inputs2 = Input(shape=(max\_length,))

se1 = Embedding(vocab\_size, 256, mask\_zero=True)(inputs2)

se2 = LSTM(256)(se1)

decoder1 = add([fe1, se2])

decoder2 = Dense(256, activation='relu')(decoder1)

outputs = Dense(vocab\_size, activation='softmax')(decoder2)

model = Model(inputs=[inputs1, inputs2], outputs=outputs)

return model

# Dummy inference function

def generate\_caption(model, tokenizer, photo):

in\_text = 'startseq'

for \_ in range(max\_length):

sequence = tokenizer.texts\_to\_sequences([in\_text])[0]

sequence = pad\_sequences([sequence], maxlen=max\_length)

yhat = model.predict([photo, sequence], verbose=0)