IMAGE CAPTIONING USING CNN + RNN

--- Python Code ---

```
import os
import numpy as np
import tensorflow as tf
from tensorflow.keras.applications.resnet50 import ResNet50, preprocess_input
from tensorflow.keras.preprocessing import image
from tensorflow.keras.models import Model
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.layers import Input, Dense, LSTM, Embedding, Dropout, add
import matplotlib.pyplot as plt
from PIL import Image

# Feature extraction using ResNet50
def extract_features(img_path):
    model = ResNet50(weights='imagenet')
    model = Model(inputs=model.inputs, outputs=model.layers[-2].output)

    img = image.load_img(img_path, target_size=(224, 224))
    x = image.img_to_array(img)
    x = np.expand_dims(x, axis=0)
    x = preprocess_input(x)

    features = model.predict(x, verbose=0)
    return features

# Dummy vocabulary
vocab = ['start', 'a', 'dog', 'on', 'the', 'beach', 'end']
word_to_index = {word: i for i, word in enumerate(vocab)}
index_to_word = {i: word for word, i in word_to_index.items()}
vocab_size = len(vocab)
max_len = 5

# Model definition
def build_model():
    inputs1 = Input(shape=(2048,))
    fe1 = Dropout(0.5)(inputs1)
    fe2 = Dense(256, activation='relu')(fe1)

    inputs2 = Input(shape=(max_len,))
    se1 = Embedding(vocab_size, 256, mask_zero=True)(inputs2)
    se2 = Dropout(0.5)(se1)
    se3 = LSTM(256)(se2)

    decoder1 = add([fe2, se3])
    decoder2 = Dense(256, activation='relu')(decoder1)
    outputs = Dense(vocab_size, activation='softmax')(decoder2)

    model = tf.keras.models.Model(inputs=[inputs1, inputs2], outputs=outputs)
    model.compile(loss='categorical_crossentropy', optimizer='adam')
    return model
```

```
# Caption generation function
def generate_caption(model, photo):
    in_text = 'start'
    for i in range(max_len):
        sequence = [word_to_index[w] for w in in_text.split() if w in word_to_index]
        sequence = pad_sequences([sequence], maxlen=max_len)
        yhat = model.predict([photo, sequence], verbose=0)
        yhat = np.argmax(yhat)
        word = index_to_word[yhat]
        if word == 'end':
            break
        in_text += ' ' + word
    return in_text.replace('start', '').strip()

# Main execution
if __name__ == "__main__":
    img_path = 'sample.jpg'
    photo = extract_features(img_path)
    model = build_model()
    caption = generate_caption(model, photo)

    img = Image.open(img_path)
    plt.imshow(img)
    plt.title("Generated Caption: " + caption)
    plt.axis('off')
    plt.show()

--- Sample Output (Simulated) ---
Generated Caption: a dog on the beach
```