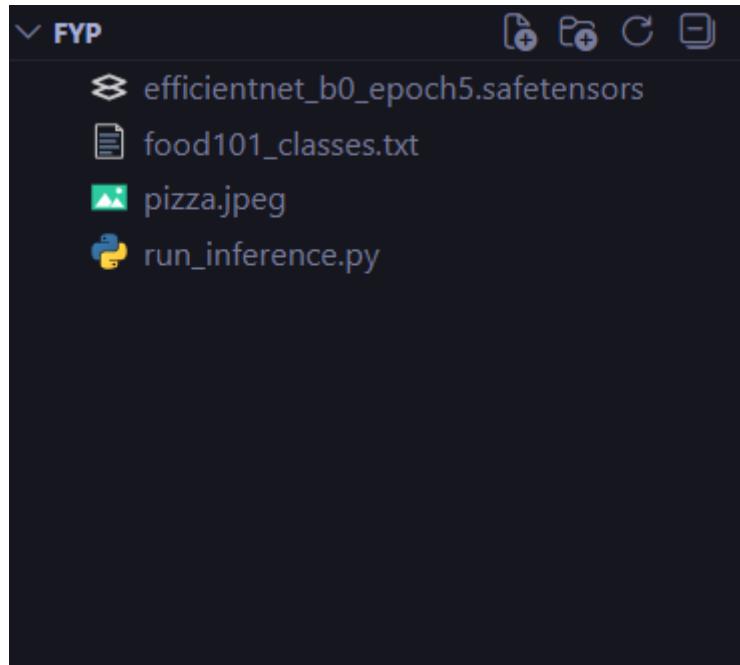


# Food Classification Inference Guide

This document explains how to run the provided PyTorch inference script using your trained **EfficientNet-B0** model saved in **SafeTensors** format.

## Project Directory Structure

Paste your directory structure here:



## Overview

This script loads a trained **EfficientNet-B0** model (saved as `efficientnet_b0_epoch5.safetensors`) and performs inference on a single image.

It outputs the predicted **Food-101 class name** and the confidence score.

## Requirements

### 1. Install Dependencies

Run the following:

```
pip install torch torchvision timm safetensors pillow
```

### 2. Required Files

Make sure the following files exist:

- efficientnet\_b0\_epoch5.safetensors → your trained model
- food101\_classes.txt → list of 101 class names (one per line)
- Test image (e.g., pizza.jpeg)

## 💡 Running Inference

### 1. Save the Script as `run_inference.py`

Use the following code:

```
import torch
from torchvision import transforms
from PIL import Image
import timm
from safetensors.torch import load_file

# --- Paths ---
model_path = "efficientnet_b0_epoch5.safetensors" # your trained model
image_path = "pizza.jpeg" # input image
classes_file = "food101_classes.txt" # classes.txt path
num_classes = 101

# --- Load model ---
model = timm.create_model('efficientnet_b0', pretrained=False,
num_classes=num_classes)
state_dict = load_file(model_path)
model.load_state_dict(state_dict)
model.eval()

# --- Preprocessing ---
preprocess = transforms.Compose([
    transforms.Resize((224, 224)),
    transforms.ToTensor(),
    transforms.Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225])
])

# --- Load class names ---
with open(classes_file) as f:
    class_names = [line.strip() for line in f]

# --- Prediction function ---
def predict(image_path):
    img = Image.open(image_path).convert("RGB")
    img_tensor = preprocess(img).unsqueeze(0) # add batch dimension
```

```
with torch.no_grad():
    outputs = model(img_tensor)
    probs = torch.softmax(outputs, dim=1)
    top1_idx = probs.argmax(dim=1).item()
    top1_prob = probs[0, top1_idx].item()

    print(f"Predicted food: {class_names[top1_idx]} ({top1_prob*100:.2f}%)")

# --- Run ---
predict(image_path)
```

## How to Run

Run the script using:

```
python run_inference.py
```

You should see an output like:

```
Predicted food: pizza(98.43%)
```