


```
from google.colab import files
uploaded = files.upload()
```

  archive.zip



- **archive.zip**(application/x-zip-compressed) - 100808871 bytes, last modified: 7/20/2025 - 100% done

Saving archive.zip to archive.zip

```
import zipfile
import os

with zipfile.ZipFile("archive.zip", 'r') as zip_ref:
    zip_ref.extractall("food101_tiny")


print(" Extracted folders:", os.listdir("food101_tiny"))
```

  Extracted folders: ['data']

```
calorie_map = {
    'apple_pie': 296,
    'burger': 295,
    'caesar_salad': 180,
    'donuts': 452,
    'edamame': 121,
    'falafel': 333,
    'french_fries': 312,
    'grilled_cheese_sandwich': 366,
    'ice_cream': 207,
    'ramen': 436
}
```


```
import os
```

```
print(os.listdir("food101_tiny"))
```

 ['data']

```
train_dir = "food101_tiny/data/train"
test_dir = "food101_tiny/data/test"
```

```
print(os.listdir("food101_tiny/data"))
```

 ['food-101-tiny']

```
train_dir = "food101_tiny/data/food-101-tiny/train"
test_dir = "food101_tiny/data/food-101-tiny/test"
```

```
import os
```

```
for root, dirs, files in os.walk("food101_tiny"):
    print(root)
```

 food101\_tiny  
 food101\_tiny/data  
 food101\_tiny/data/food-101-tiny  
 food101\_tiny/data/food-101-tiny/valid  
 food101\_tiny/data/food-101-tiny/valid/cannoli  
 food101\_tiny/data/food-101-tiny/valid/apple\_pie  
 food101\_tiny/data/food-101-tiny/valid/french\_toast  
 food101\_tiny/data/food-101-tiny/valid/bibimbap  
 food101\_tiny/data/food-101-tiny/valid/sushi  
 food101\_tiny/data/food-101-tiny/valid/ramen  
 food101\_tiny/data/food-101-tiny/valid/ice\_cream

```

food101_tiny/data/food-101-tiny/valid/tiramisu
food101_tiny/data/food-101-tiny/valid/edamame
food101_tiny/data/food-101-tiny/valid/falafel
food101_tiny/data/food-101-tiny/train
food101_tiny/data/food-101-tiny/train/cannoli
food101_tiny/data/food-101-tiny/train/apple_pie
food101_tiny/data/food-101-tiny/train/french_toast
food101_tiny/data/food-101-tiny/train/bibimbap
food101_tiny/data/food-101-tiny/train/sushi
food101_tiny/data/food-101-tiny/train/ramen
food101_tiny/data/food-101-tiny/train/ice_cream
food101_tiny/data/food-101-tiny/train/tiramisu
food101_tiny/data/food-101-tiny/train/edamame
food101_tiny/data/food-101-tiny/train/falafel

```

```

train_dir = "food101_tiny/data/food-101-tiny/train"
val_dir   = "food101_tiny/data/food-101-tiny/valid"

```

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

```

img_size = (224, 224)
batch_size = 16

```

```
datagen = ImageDataGenerator(rescale=1./255)
```

```

train_gen = datagen.flow_from_directory(
    train_dir,
    target_size=img_size,
    batch_size=batch_size,
    class_mode='categorical'
)

```

```

val_gen = datagen.flow_from_directory(
    val_dir,
    target_size=img_size,
    batch_size=batch_size,
    class_mode='categorical'
)

```

➡ Found 1500 images belonging to 10 classes.  
Found 500 images belonging to 10 classes.

```

# Model building
import tensorflow as tf

base_model = tf.keras.applications.MobileNetV2(
    input_shape=(224, 224, 3), include_top=False, weights='imagenet'
)
base_model.trainable = False

```

```

model = tf.keras.Sequential([
    base_model,
    tf.keras.layers.GlobalAveragePooling2D(),
    tf.keras.layers.Dense(128, activation='relu'),
    tf.keras.layers.Dropout(0.3),
    tf.keras.layers.Dense(10, activation='softmax') # 10 classes
])

```

```
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
```

```

# Train
model.fit(train_gen, validation_data=val_gen, epochs=5)

```

➡ Downloading data from [https://storage.googleapis.com/tensorflow/keras-applications/mobilenet\\_v2/mobilenet\\_v2\\_weights\\_tf\\_dim\\_ordering\\_tf\\_9406464/9406464](https://storage.googleapis.com/tensorflow/keras-applications/mobilenet_v2/mobilenet_v2_weights_tf_dim_ordering_tf_9406464/9406464) 1s 0us/step  
/usr/local/lib/python3.11/dist-packages/keras/src/trainers/data\_adapters/py\_dataset\_adapter.py:121: UserWarning: Your `PyDataset` class  
self.\_warn\_if\_super\_not\_called()  
Epoch 1/5  
94/94 ————— 88s 866ms/step - accuracy: 0.3696 - loss: 1.8371 - val\_accuracy: 0.7400 - val\_loss: 0.7885  
Epoch 2/5  
94/94 ————— 78s 836ms/step - accuracy: 0.7529 - loss: 0.7229 - val\_accuracy: 0.7540 - val\_loss: 0.7536  
Epoch 3/5  
94/94 ————— 80s 852ms/step - accuracy: 0.8509 - loss: 0.4958 - val\_accuracy: 0.7520 - val\_loss: 0.7888  
Epoch 4/5

94/94 ————— 81s 860ms/step - accuracy: 0.8734 - loss: 0.4279 - val\_accuracy: 0.7780 - val\_loss: 0.6642  
Epoch 5/5  
94/94 ————— 78s 829ms/step - accuracy: 0.9156 - loss: 0.2830 - val\_accuracy: 0.7620 - val\_loss: 0.7071  
<keras.src.callbacks.history.History at 0x7adf39f0b810>

```
import numpy as np
from tensorflow.keras.utils import load_img, img_to_array

def predict_food(image_path):
    img = load_img(image_path, target_size=(224, 224))
    img_array = img_to_array(img) / 255.0
    img_array = np.expand_dims(img_array, axis=0)

    pred = model.predict(img_array)
    idx = np.argmax(pred)
    class_name = list(train_gen.class_indices.keys())[idx]
    calories = calorie_map.get(class_name, "Unknown")

    print(f"🍽️ Predicted Food: {class_name}")
    print(f"🔥 Estimated Calories: {calories} kcal")
```

```
from google.colab import files
uploaded = files.upload()
```

📁 Choose Files falafel.jpg

- **falafel.jpg**(image/jpeg) - 60668 bytes, last modified: 7/21/2025 - 100% done  
Saving falafel.jpg to falafel.jpg

```
predict_food("falafel.jpg")
```

📁 1/1 ————— 0s 85ms/step

🍽️ Predicted Food: falafel  
🔥 Estimated Calories: 333 kcal