

```

import tensorflow as tf

from tensorflow.keras.applications import MobileNetV2

from tensorflow.keras.applications.mobilenet_v2 import preprocess_input, decode_predictions

from tensorflow.keras.preprocessing import image

import numpy as np


# Load the MobileNetV2 model pre-trained on ImageNet data
model = MobileNetV2(weights='imagenet')


# Load and preprocess an image for prediction
img_path = 'path_to_your_image.jpg' # Replace with the path to your image
img = image.load_img(img_path, target_size=(224, 224))
img = image.img_to_array(img)
img = np.expand_dims(img, axis=0)
img = preprocess_input(img)


# Make predictions
predictions = model.predict(img)


# Decode and print the top 5 predicted classes
decoded_predictions = decode_predictions(predictions, top=5)[0]


for i, (imagenet_id, label, score) in enumerate(decoded_predictions):
    print(f"{i + 1}: {label} ({score:.2f})")

from flask import Flask, request, jsonify

from chatterbot import ChatBot

from chatterbot.trainers import ChatterBotCorpusTrainer


app = Flask(__name)

```

```
# Create a chatbot instance

chatbot = ChatBot('MyChatbot')

trainer = ChatterBotCorpusTrainer(chatbot)


# Train the chatbot on English language data
trainer.train('chatterbot.corpus.english')


@app.route('/ask', methods=['POST'])
def ask():
    user_message = request.json['message']
    response = chatbot.get_response(user_message)
    return jsonify({'response': str(response)})


if __name__ == '__main__':
    app.run(debug=True)
```