```
from flask import Flask, request, jsonify
from tensorflow.keras.models import
load_model
from
tensorflow.keras.applications.resnet50
import preprocess_input
from
tensorflow.keras.preprocessing.image
import img_to_array, load_img
import numpy as np
from PIL import Image
import io

app = Flask(__name__)

# Load the pre-trained model (assumed to
be trained on blood cell dataset)
model =
load_model('blood_cell_model.h5')  #
Replace with your model path
class_names = ['Neutrophil', 'Eosinophil',
```

```
'Lymphocyte', 'Monocyte', 'Basophil',
                'Immature Granulocyte',
'Erythroblast', 'Platelet']  # Adjust based on
your model

def preprocess_image(image):
    # Resize image to match model input
(e.g., 224x224 for ResNet50)
    image = image.resize((224, 224))
    image = img_to_array(image)
    image = np.expand_dims(image,
axis=0)
    image = preprocess_input(image)
    return image

@app.route('/classify', methods=['POST'])
def classify():
    if 'image' not in request.files:
        return jsonify({'error': 'No image
uploaded'}), 400
```

```
    file = request.files['image']
    try:
        # Load and preprocess image
        image =
Image.open(io.BytesIO(file.read()))
        if image.mode != 'RGB':
            image = image.convert('RGB')
        image = preprocess_image(image)

        # Predict
        predictions = model.predict(image)
        predicted_class =
np.argmax(predictions[0])
        confidence = float(predictions[0]
[predicted_class])

        return jsonify({
            'class':
class_names[predicted_class],
            'confidence': confidence
        })
```

```
    except Exception as e:
        return jsonify({'error': str(e)}), 500

if __name__ == '__main__':
    app.run(debug=True, host='0.0.0.0',
port=5000)
```