Use of AI

[1]. Tool: ChatGPT

Prompt:
# POST Request @app.route('/inference', methods=['POST']) def inference(): global model if model is None: load_kmodel() file = None if 'file' in request.files: file = request.files['file'] elif 'image' in request.files: file = request.files['image'] elif request.data: file = io.BytesIO(request.data) else: return jsonify({"error": "No image provided"}), 400 img = Image.open(file) processed = preprocess_image(img) preds = model.predict(processed)

if i have all of this in my inference request, how do i actually use the preds value to calculate whether my image is damaged or not.

Output:

Perfect question — you’re right at the final step 🎯

Once you have `preds = model.predict(processed)`, that gives you the **raw output** from your model (logits or probabilities).

Now, how you interpret that depends on your model’s output layer — let’s walk through the two most common cases 👇

🧠 Case 1: Binary Classification (1 output neuron with sigmoid)

If your model’s final layer is something like:

```
Dense(1, activation='sigmoid')
```

then `preds` will look like:

```
[[0.82]]
```

That means your model predicts “damage” with 82% probability.
You can do this:

```
prob = float(preds[0][0])  # Extract scalar
cls = 1 if prob >= 0.5 else 0  # 1 = damage, 0 = no damage

label = "damage" if cls == 1 else "no_damage"
return jsonify({"prediction": label, "confidence": prob}), 200
```

🧠 Case 2: Two-Class Softmax (2 output neurons)

If your final layer is:

```
Dense(2, activation='softmax')
```

then `preds` will look like:

```
[[0.25, 0.75]]
```

— where the first number is probability of class 0, and the second for class 1.

You can decode it like this:

```
cls = int(np.argmax(preds, axis=1)[0])
confidence = float(np.max(preds))

# Assuming class 0 = damage, class 1 = no_damage
label = "damage" if cls == 0 else "no_damage"
return jsonify({"prediction": label, "confidence": confidence}), 200
```

✅ Full Updated `/inference` Endpoint

Here’s a full working version that handles both cases safely:

```
@app.route('/inference', methods=['POST'])
def inference():
    global model
    if model is None:
        load_kmodel()

    file = None
    if 'file' in request.files:
        file = request.files['file']
    elif 'image' in request.files:
        file = request.files['image']
    elif request.data:
        file = io.BytesIO(request.data)
    else:
        return jsonify({"error": "No image provided"}), 400

    # Load and preprocess image
    img = Image.open(file)
    processed = preprocess_image(img)

    # Run model prediction
    preds = model.predict(processed)

    # Handle sigmoid or softmax outputs
    if preds.shape[1] == 1:  # Binary sigmoid
```

```
        prob = float(preds[0][0])
        cls = 1 if prob >= 0.5 else 0
        label = "damage" if cls == 1 else "no_damage"
        confidence = prob if cls == 1 else 1 - prob
    else:  # Softmax (2 outputs)
        cls = int(np.argmax(preds, axis=1)[0])
        confidence = float(np.max(preds))
        label = "damage" if cls == 0 else "no_damage"

    return jsonify({"prediction": label, "confidence": confidence}), 200
```

💡 Tip: You can quickly test it with:

```
curl -X POST -F "file=@test_damage.jpg" http://localhost:5000/inference
```

Would you like me to check how to automatically detect whether your model uses sigmoid or softmax from the `.h5` file?