**EMOTION DETECTION FROM UPLOADED IMAGES**

**PROBLEM STATEMENT:**

To develop and design, implement, and optimize a complete solution that integrates machine learning, computer vision, and user interface design to classify emotions of uploaded images.

**OBJECTIVE:**

The goal of this project is to develop a **Streamlit-based application** that enables users to upload an image, which will then be processed to detect the **emotion** of the person in the image using **Convolutional Neural Networks (CNNs)**.

**BUSINESS USE CASES:**

* Healthcare: Mental Health Monitoring and Support
* Education: Personalized Learning and Engagement
* Customer Service: Enhancing User Experience
* Market Research: Understanding Consumer Sentiment
* Human Resources: Improving Employee Engagement and Well-being

**METHODOLOGY, MODELS USED, AND EVALUATION RESULTS:**

**Step1: CNN Model Development**

* **Datasets:** [FER-2013](https://www.kaggle.com/datasets/damnithurts/fer2013-dataset-images)

This image dataset contains images that fall into 7 categories 'Angry', 'Disgust', 'Fear', 'Happy', 'Sad', 'Surprise', 'Neutral'.

* **Facial Feature Extraction**

Dlib is used to detect facial landmarks (68 points) which highlights key features (e.g., eyes, mouth, eyebrows) that contribute to emotion detection. These landmark-based feature engineering can be experiment to extract more useful information for emotion classification.

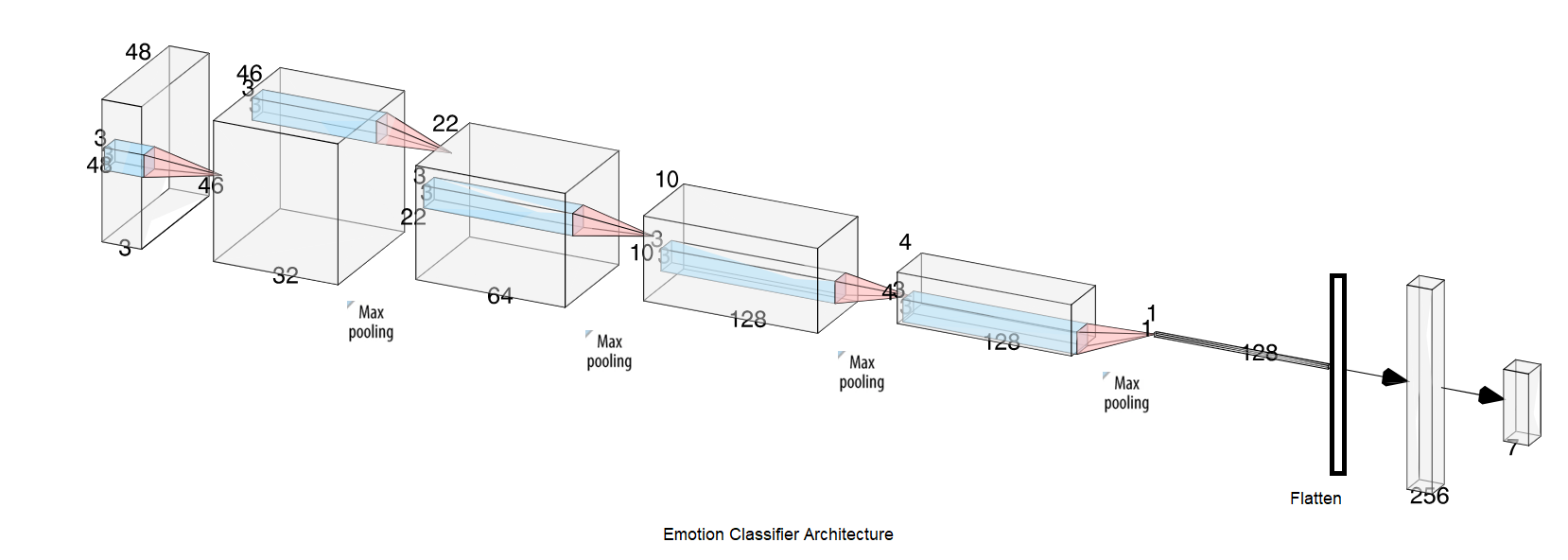
* **Models**
* **Emotion Classifier – Custom Model**

**Explanation of the Layers:**

* Conv Layers: These are convolutional layers that extract features from the input image. Each layer has a specific number of filters (e.g., 32, 64, 128), which helps capture various visual patterns.
* MaxPooling Layers: Pooling layers reduce the spatial dimensions of the image, which helps in reducing the computational load and preventing overfitting. Here, we use a 2x2 MaxPooling operation.
* Flatten: After the convolution and pooling operations, the multi-dimensional data is flattened into a 1D vector to feed into the fully connected layers.
* Fully Connected Layers (fc1, fc2): These layers act as a traditional neural network that connects all the neurons in the previous layers to produce the final output. fc2 has 7 output neurons corresponding to the 7 emotion classes.
* Dropout: Dropout is used to randomly disable a fraction of neurons during training to avoid overfitting.

**High-Level Workflow:**

* Input Image: The image (RGB, 3 channels) is passed through the model.
* 5 Convolutional layers extract features, progressively increasing the number of channels.
* Pooling layers reduce the spatial size.
* Flatten: Flatten the output of the last pooling layer into a 1D vector.
* 2 Fully connected layers produce a final vector of size 7, each corresponding to a different emotion.
* Output: The model outputs the predicted emotion class.

****

 **Input Layer**: 48x48x3 (RGB image).

 **Conv1**: Convolution (32 filters of 3x3), output size 46x46x32.

 **Conv2**: Convolution (64 filters of 3x3), output size 44x44x64.

 **Pool1**: Max Pooling (2x2), output size 22x22x64.

 **Conv3**: Convolution (128 filters of 3x3), output size 20x20x128.

 **Pool2**: Max Pooling (2x2), output size 10x10x128.

 **Conv4**: Convolution (128 filters of 3x3), output size 8x8x128.

 **Pool3**: Max Pooling (2x2), output size 4x4x128.

 **Conv5**: Convolution (128 filters of 3x3), output size 2x2x128.

 **Pool4**: Max Pooling (2x2), output size 1x1x128.

 **Flatten**: Flattening the output, 128.

 **FC1**: Fully Connected (256 units), output size 256.

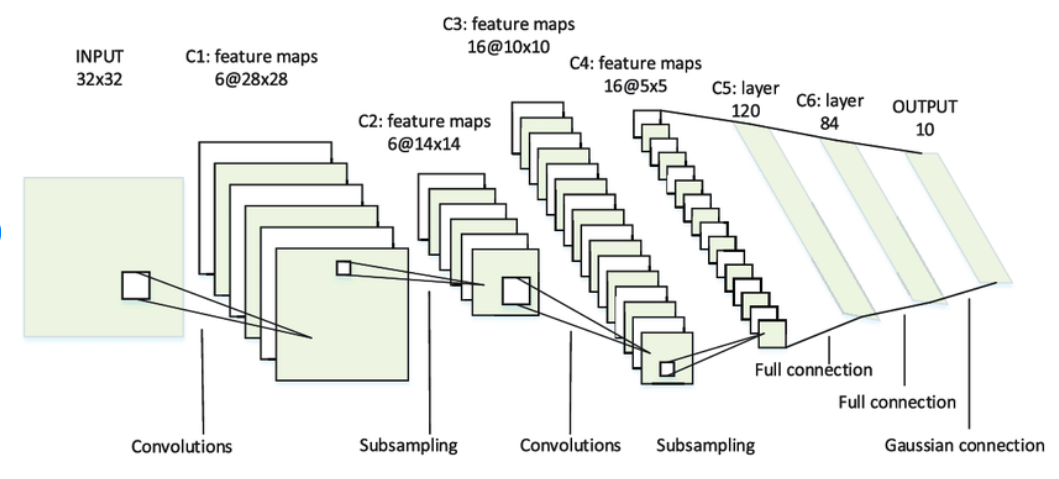
 **Dropout**: Dropout (50%).

 **FC2**: Fully Connected (7 units), output size 7 (emotion classes).

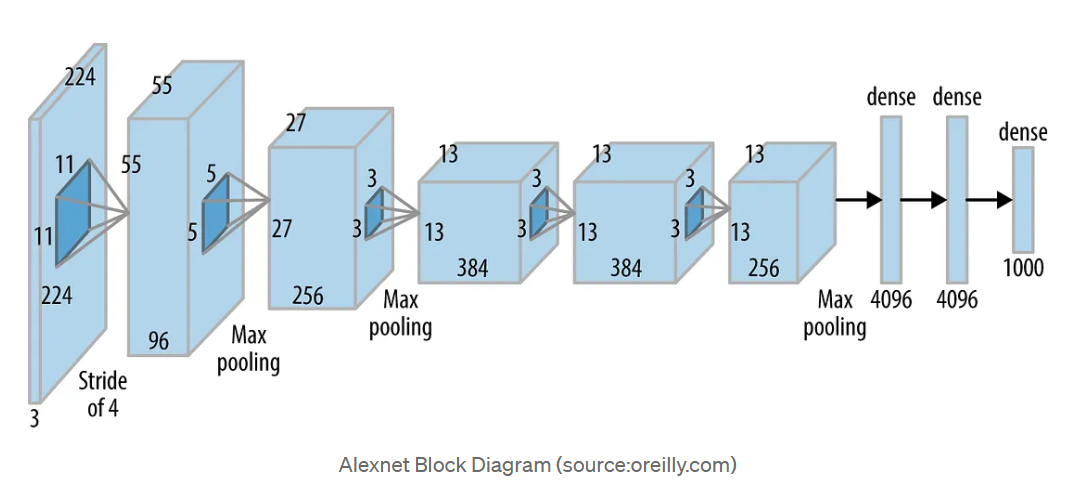
* **LeNet**

LeNet Summary in Terms of Layers:

* **Input Layer**: 32x32x1 (grayscale image).
* **C1**: Convolution (6 filters of 5x5), output size 28x28x6.
* **S2**: Average Pooling (2x2), output size 14x14x6.
* **C3**: Convolution (16 filters of 5x5), output size 10x10x16.
* **S4**: Average Pooling (2x2), output size 5x5x16.
* **C5**: Fully Connected (120 units), output size 1x120.
* **F6**: Fully Connected (84 units), output size 1x84.
* **Output**: Fully Connected (10 units for classes 0-9), output size 1x10.



* **Alex Net**

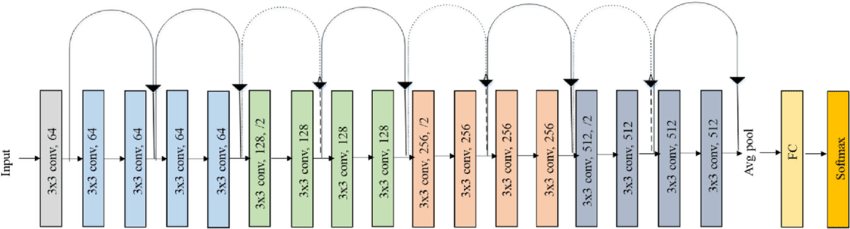
****

Summary of AlexNet Layers:

* **Input Layer**: 224x224x3 (RGB image).
* **Conv1**: Convolution (96 filters of 11x11), output size 55x55x96, ReLU, LRN, Max Pooling.
* **Conv2**: Convolution (256 filters of 5x5), output size 27x27x256, ReLU, LRN, Max Pooling.
* **Conv3**: Convolution (384 filters of 3x3), output size 13x13x384, ReLU.
* **Conv4**: Convolution (384 filters of 3x3), output size 13x13x384, ReLU.
* **Conv5**: Convolution (256 filters of 3x3), output size 13x13x256, ReLU, Max Pooling.
* **Flatten**: Flattening the output of Conv5, 9216 units.
* **FC1**: Fully Connected (4096 units), ReLU.
* **Dropout**: Dropout (50%).
* **FC2**: Fully Connected (4096 units), ReLU.
* **FC3**: Fully Connected (1000 units), output size 1000 (for 1000 classes).
* **ResNet18**

Summary of ResNet-18 Layers:

* **Input**: 224x224x3 (RGB image).
* **Conv1**: Convolution (64 filters of 7x7), ReLU, Batch Norm, Max Pooling → 112x112x64.
* **Stage 1**: 2 residual blocks with 64 filters (3x3 Conv, ReLU, Batch Norm).
* **Stage 2**: 2 residual blocks with 128 filters (3x3 Conv, ReLU, Batch Norm).
* **Stage 3**: 2 residual blocks with 256 filters (3x3 Conv, ReLU, Batch Norm).
* **Stage 4**: 2 residual blocks with 512 filters (3x3 Conv, ReLU, Batch Norm).
* **Global Average Pooling** → 512.
* **Fully Connected Layer (FC)** → 1000 units (for classification).
* **Output**: Softmax (1000 class probabilities).



Initially, use pre-trained models (such as ResNet-18, AlexNet, VGG, etc.) with the output layer set to 7 neurons to classify one of the 7 emotions (Anger, Disgust, Fear, Happy, Sad, Surprise, Neutral) along with the custom architecture. The main steps include **image preprocessing**, **emotion classification**, and **evaluation** (accuracy, precision, recall, f1 score). Later, evaluate the system by incorporating **facial landmarks** to improve the emotion classification performance. All the model training are done for 5 epochs.

|  |  |  |
| --- | --- | --- |
| Models | Metrics (without facial landmarks) | Metrics (with facial landmarks) |
| EmotionClassifier | \*\*\*\*\*\*\*Train\*\*\*\*\*\*\*\*\*\*  Accuracy score : 0.5566198753004284  Precision : 0.513464447777855  Recall : 0.47028281485820916  F1 score: 0.47170729424017793  \*\*\*\*\*\*\*Test\*\*\*\*\*\*\*\*\*\*  Accuracy score : 0.5351072722206743  Precision : 0.5275499080699518  Recall : 0.45569405182347245  F1 score: 0.46146266114609974 | \*\*\*\*\*\*\*Train\*\*\*\*\*\*\*\*\*\*  Accuracy score : 0.6006478804556062  Precision : 0.568781534836548  Recall : 0.5282618945130407  F1 score: 0.5381434297780456  \*\*\*\*\*\*\*Test\*\*\*\*\*\*\*\*\*\*  Accuracy score : 0.5572582892170521  Precision : 0.5597644109451009  Recall : 0.5029124983240104  F1 score: 0.5116266177665098 |
| LeNet | \*\*\*\*\*\*\*Train\*\*\*\*\*\*\*\*\*\*  Accuracy score : 0.4199379985370441  Precision : 0.39640880018709185  Recall : 0.33793801021291714  F1 score: 0.32902585995181965  \*\*\*\*\*\*\*Test\*\*\*\*\*\*\*\*\*\*  Accuracy score : 0.41083867372527166  Precision : 0.3224850276736119  Recall : 0.3298032411184854  F1 score: 0.31660582691150657 | \*\*\*\*\*\*\*Train\*\*\*\*\*\*\*\*\*\*  Accuracy score : 0.4088961649656902  Precision : 0.4497167397038141  Recall : 0.3283923955052657  F1 score: 0.32018965459791815  \*\*\*\*\*\*\*Test\*\*\*\*\*\*\*\*\*\*  Accuracy score : 0.3921705210365004  Precision : 0.3181445126953924  Recall : 0.3354178266215252  F1 score: 0.32019570094764704 |
| AlexNet | \*\*\*\*\*\*\*Train\*\*\*\*\*\*\*\*\*\*  Accuracy score : 0.2513149186666202  Precision : 0.0359021312380886  Recall : 0.14285714285714285  F1 score: 0.057383046749487014  \*\*\*\*\*\*\*Test\*\*\*\*\*\*\*\*\*\*  Accuracy score : 0.24714405126776262  Precision : 0.0353062930382518  Recall : 0.14285714285714285  F1 score: 0.0566194306140687 | \*\*\*\*\*\*\*Train\*\*\*\*\*\*\*\*\*\*  Accuracy score : 0.37347173360270297  Precision : 0.3303206570446823  Recall : 0.3157385728670765  F1 score: 0.31719534732534477  \*\*\*\*\*\*\*Test\*\*\*\*\*\*\*\*\*\*  Accuracy score : 0.4197548063527445  Precision : 0.42654661071716926  Recall : 0.3632263000853471  F1 score: 0.3500067350090822 |
| ResNet | \*\*\*\*\*\*\*Train\*\*\*\*\*\*\*\*\*\*  Accuracy score : 0.4389912570970776  Precision : 0.43103645583647593  Recall : 0.3705102442508971  F1 score: 0.3730503477496483  \*\*\*\*\*\*\*Test\*\*\*\*\*\*\*\*\*\*  Accuracy score : 0.4210086375034829  Precision : 0.4209445948541482  Recall : 0.36252754487110384  F1 score: 0.33802640365436615 | \*\*\*\*\*\*\*Train\*\*\*\*\*\*\*\*\*\*  Accuracy score : 0.4319899683026229  Precision : 0.4045164001337909  Recall : 0.36330065915867876  F1 score: 0.36508190636126436  \*\*\*\*\*\*\*Test\*\*\*\*\*\*\*\*\*\*  Accuracy score : 0.4254667038172193  Precision : 0.4326521068396419  Recall : 0.3658807807641145  F1 score: 0.36819386856372727 |

**The EmotionClassifier model architecture and weights are saved.**

EmotionClassifier(

(conv1): Conv2d(3, 32, kernel\_size=(3, 3), stride=(1, 1))

(conv2): Conv2d(32, 64, kernel\_size=(3, 3), stride=(1, 1))

(conv3): Conv2d(64, 128, kernel\_size=(3, 3), stride=(1, 1))

(conv4): Conv2d(128, 128, kernel\_size=(3, 3), stride=(1, 1))

(conv5): Conv2d(128, 128, kernel\_size=(3, 3), stride=(1, 1))

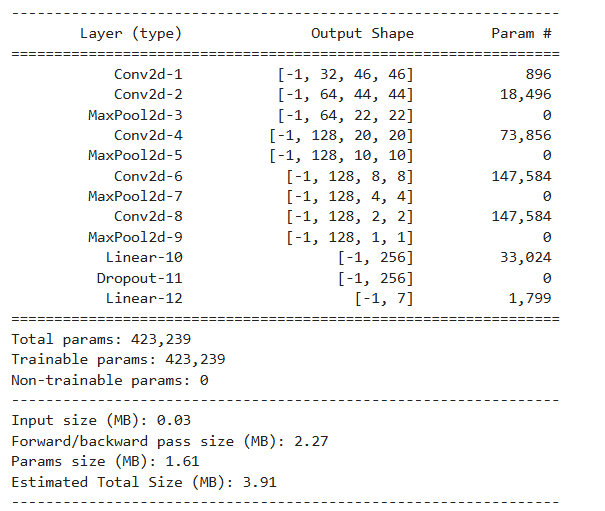
(pool): MaxPool2d(kernel\_size=2, stride=2, padding=0, dilation=1, ceil\_mode=False)

(dropout): Dropout(p=0.5, inplace=False)

(fc1): Linear(in\_features=128, out\_features=256, bias=True)

(fc2): Linear(in\_features=256, out\_features=7, bias=True)

)

****

**Step2: Streamlit User interface**

The **emotion detection system** using a **Streamlit** web app that allows users to upload an image, process it using a **CNN model** trained for emotion classification, and display the predicted emotion. Additionally, the system incorporates **facial landmark detection** to potentially enhance the classification results.

 **Model Definition (EmotionClassifier class)**:

* Defines a **Convolutional Neural Network (CNN)** for emotion classification with 5 convolutional layers and 2 fully connected layers.The network accepts RGB images and outputs a probability distribution over 7 emotions (Angry, Disgust, Fear, Happy, Sad, Surprise, Neutral).

 **Pre-trained Model Loading**:

* Loads a pre-trained emotion detection model (model\_complete.pth) that has been trained to classify emotions from facial images and it set to **evaluation mode** using model.eval() for inference.

 **Dlib's Face Detection**:

* Uses **Dlib**'s **frontal face detector** and **68-point facial landmark predictor** to locate faces and extract key facial features.If faces are detected , then these landmarks are drawn on the image to visualize where the system detects facial features.

 **Image Preprocessing**:

* The uploaded image is converted to RGB and then resized to (128, 128) for face detection, later resized to (48, 48) for feeding it into the CNN model.
* The image is transformed into a **tensor** (required for PyTorch model input) using transforms.Resize and transforms.ToTensor().

 **User Interface (Streamlit)**:

* The user uploads an image via the **Streamlit sidebar** using the file\_uploader widget.If the image is successfully uploaded, it is displayed to the user with the st.image() function. Only jpeg,jpg, png image files are allowed to be uploaded else the file\_uploader prompts an error message.

 **Face Detection and Error Handling**:

* The image is processed to detect faces using **Dlib's detector**. If no faces are detected:
  + An error message is displayed to the user asking whether to continue without detected faces.
  + The user is provided two buttons: "Clear" (to reload the file uploader) and "Continue" (to proceed with emotion prediction despite no faces).
* If **multiple faces** are detected, a message indicating multiple faces is displayed on the sidebar.

 **Landmark Detection and Display**:

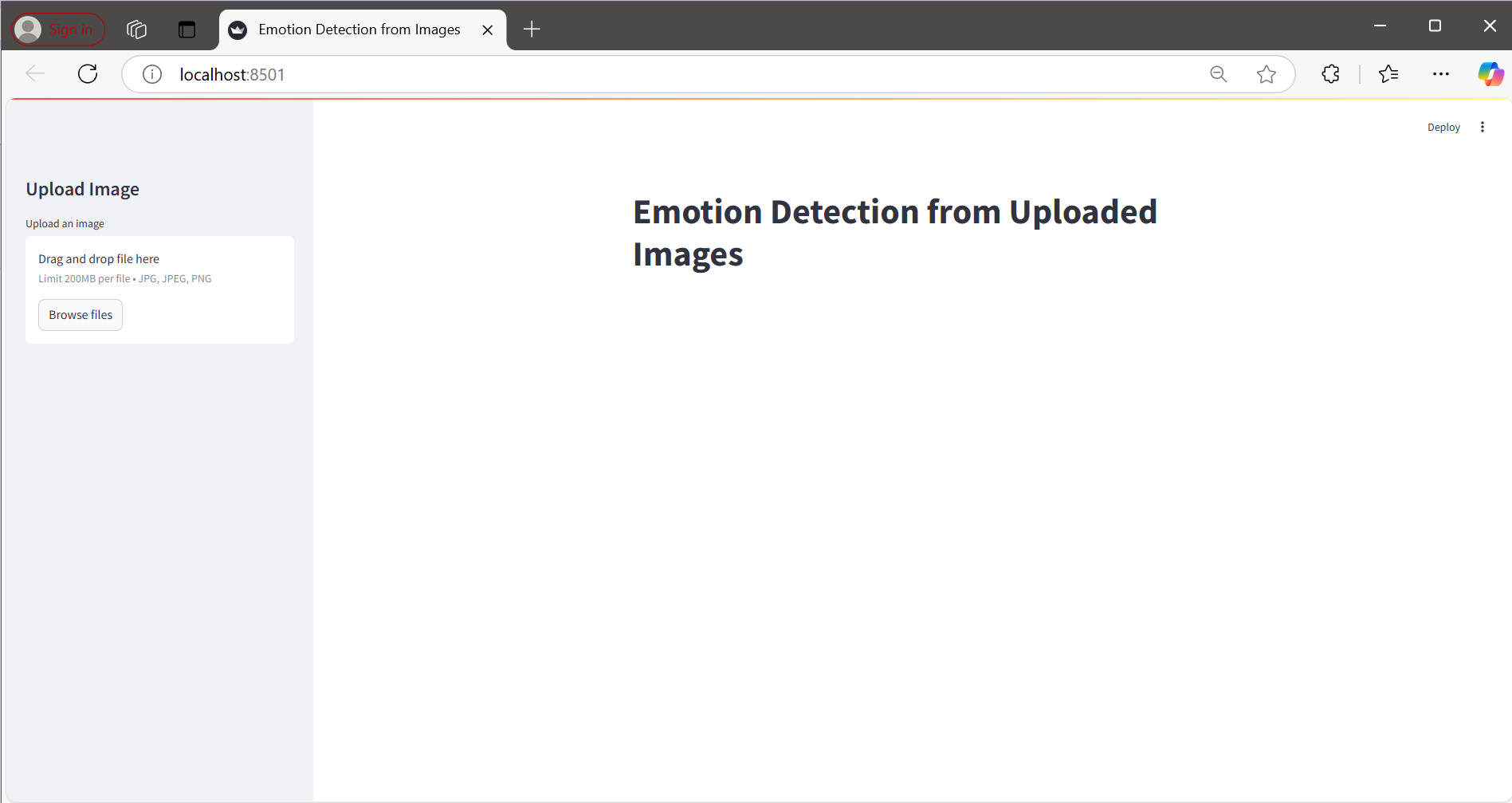
* If faces are detected, **68 facial landmarks** are extracted and visualized on the image using **OpenCV** (cv2.circle to mark the points).
* This information can be used to enhance emotion classification or just for visual confirmation.

 **Emotion Prediction**:

* The processed image is passed through the **pre-trained emotion classification model** to predict the emotion.
* The prediction is made using torch.no\_grad() to prevent tracking gradients (since we’re not training the model during inference).
* The predicted emotion label is displayed on the webpage using **Streamlit.**

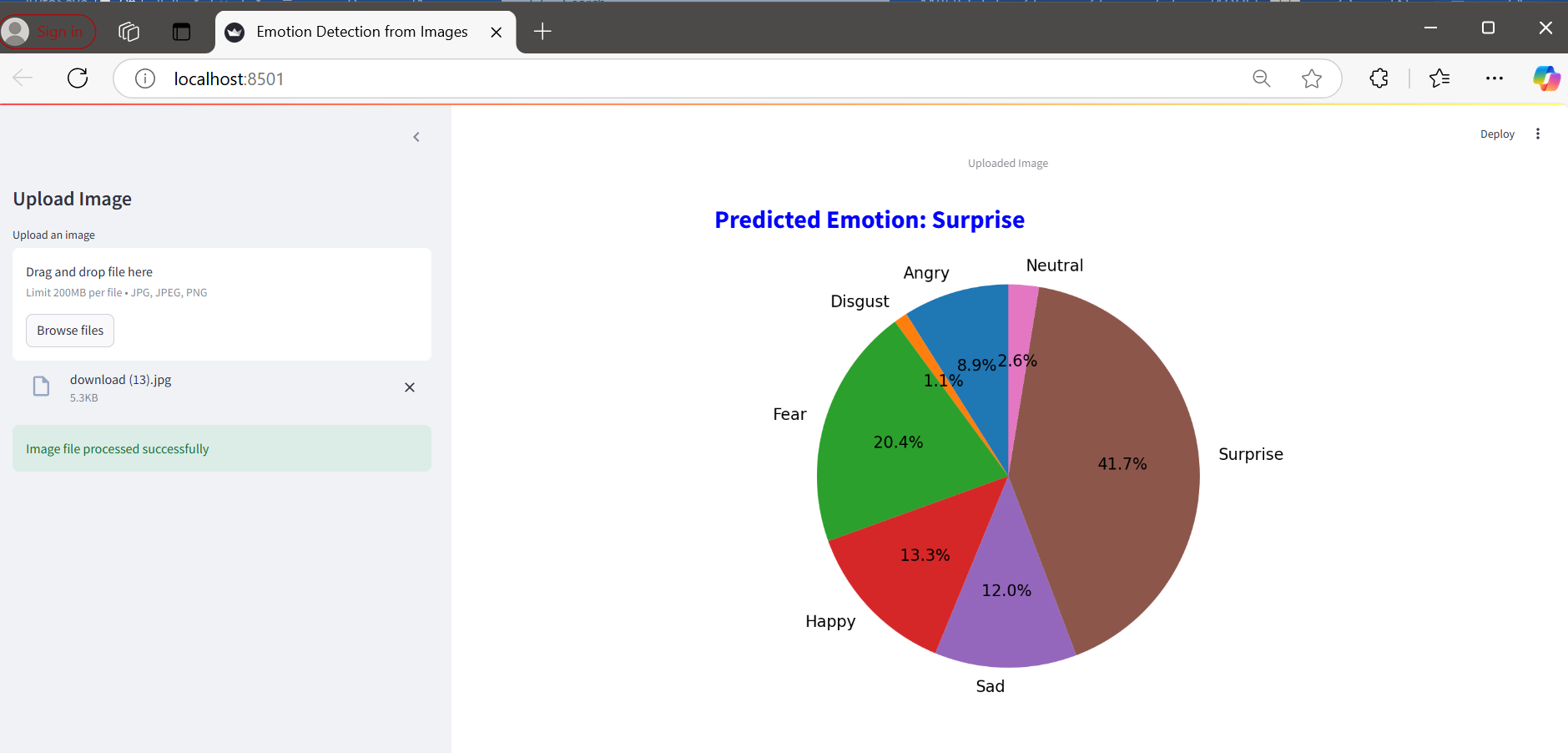
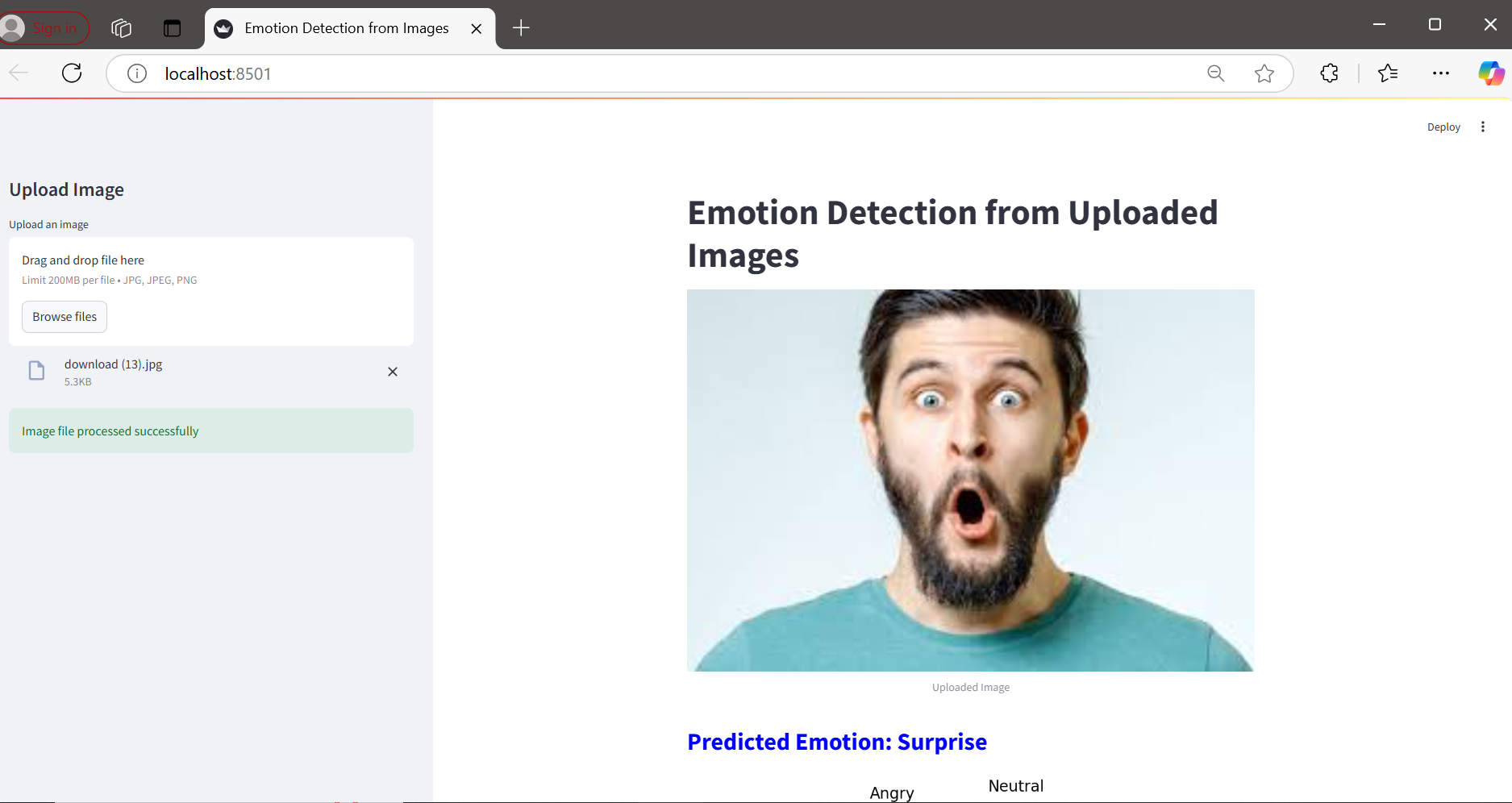
 **Prediction Visualization**:

* A **pie chart** is generated using **Matplotlib** to show the **probability distribution** over the 7 possible emotions, allowing the user to see the confidence level of each predicted emotion.

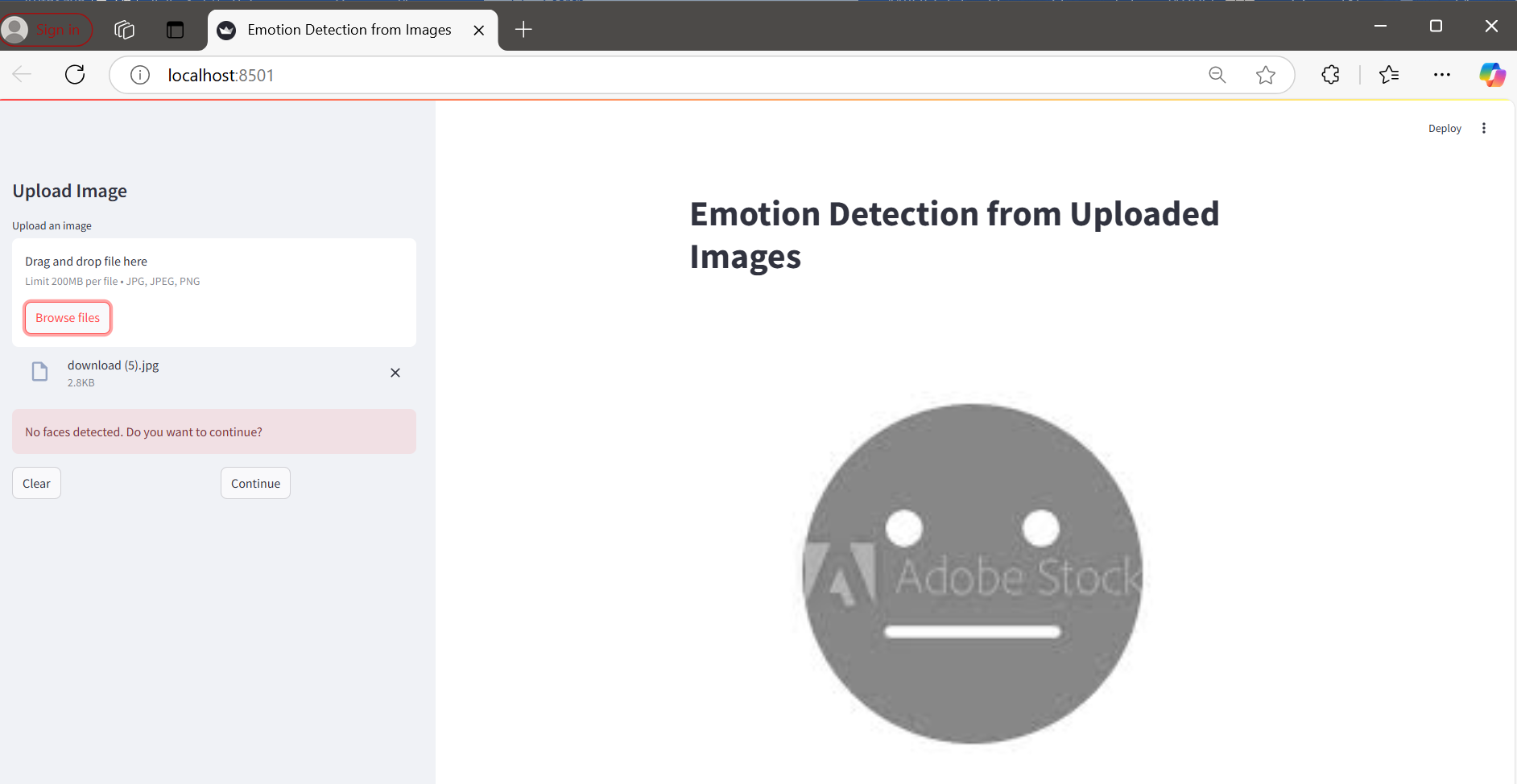


Streamlit UI

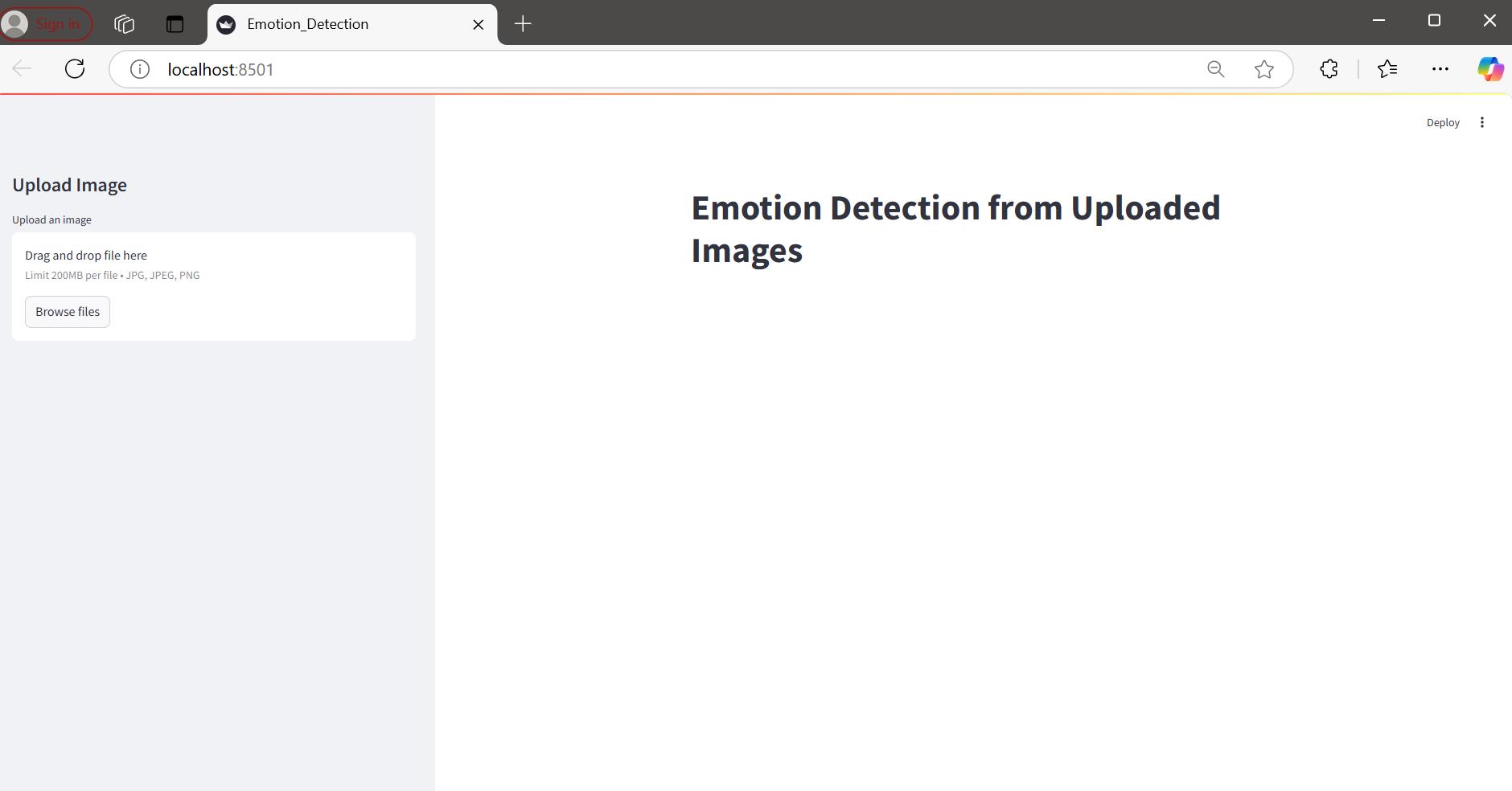
Case 1: One face detected and corresponding output



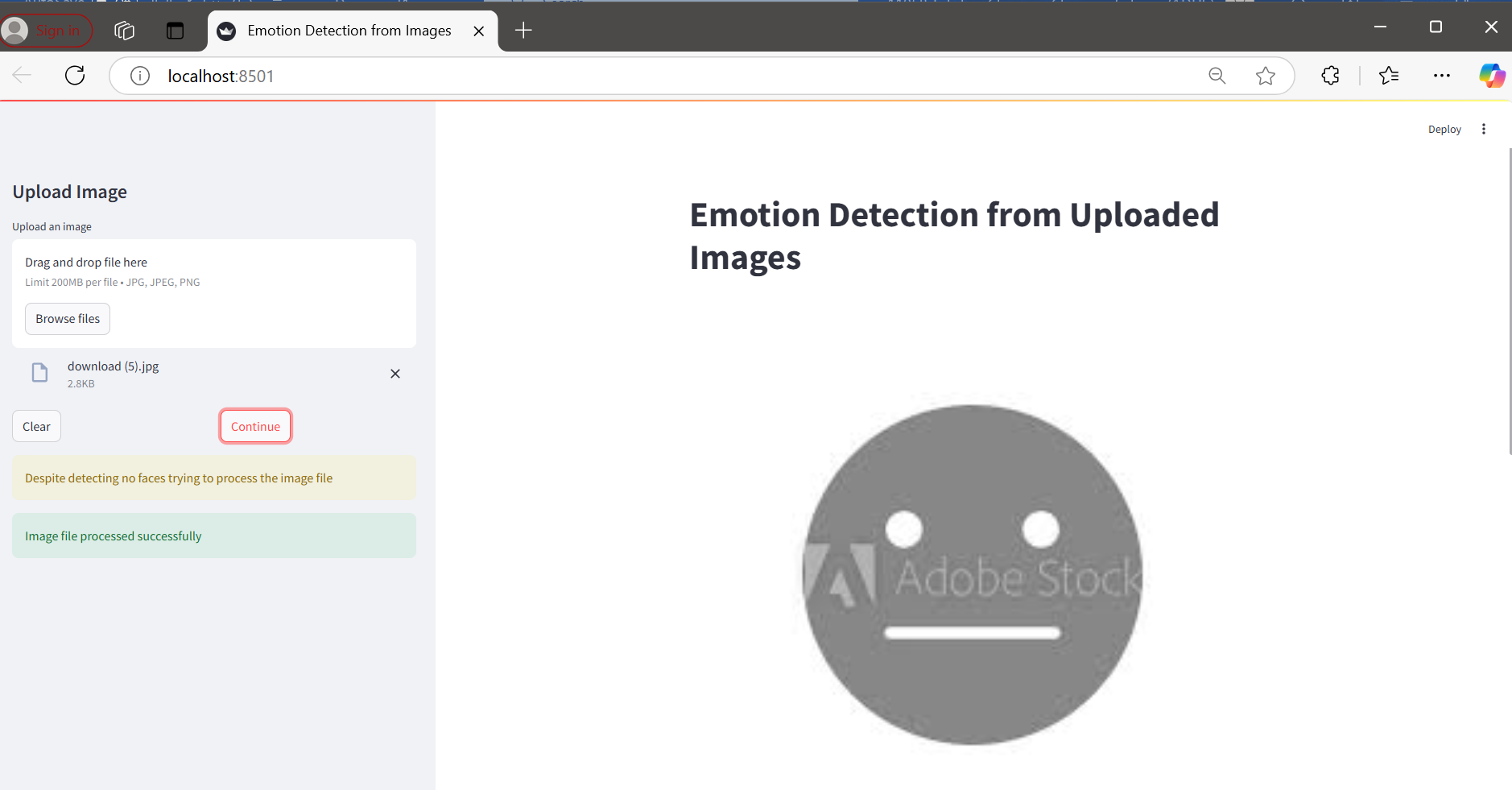
Case 2: No faces detected

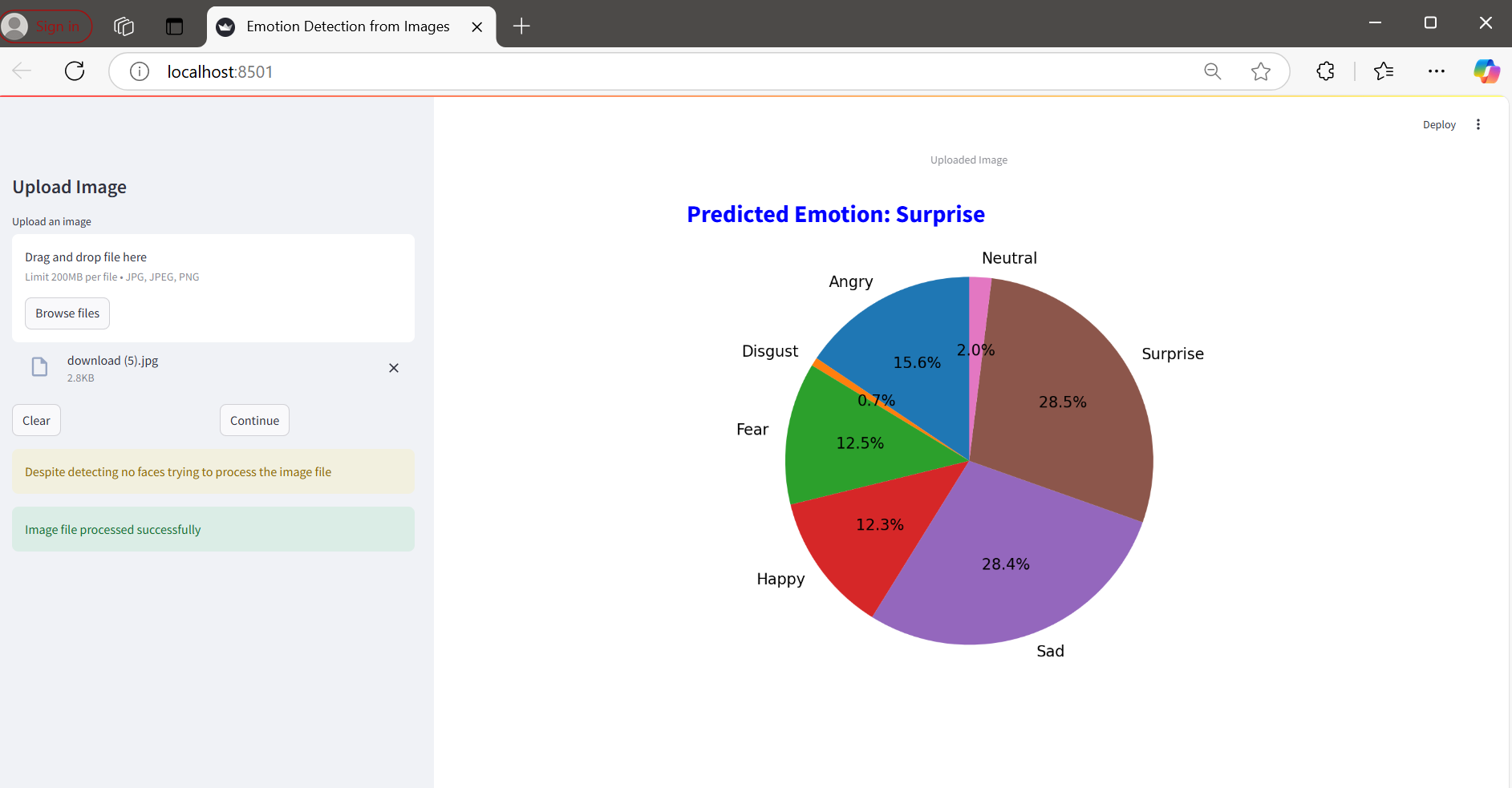


Case 2 a: User chooses the clear button

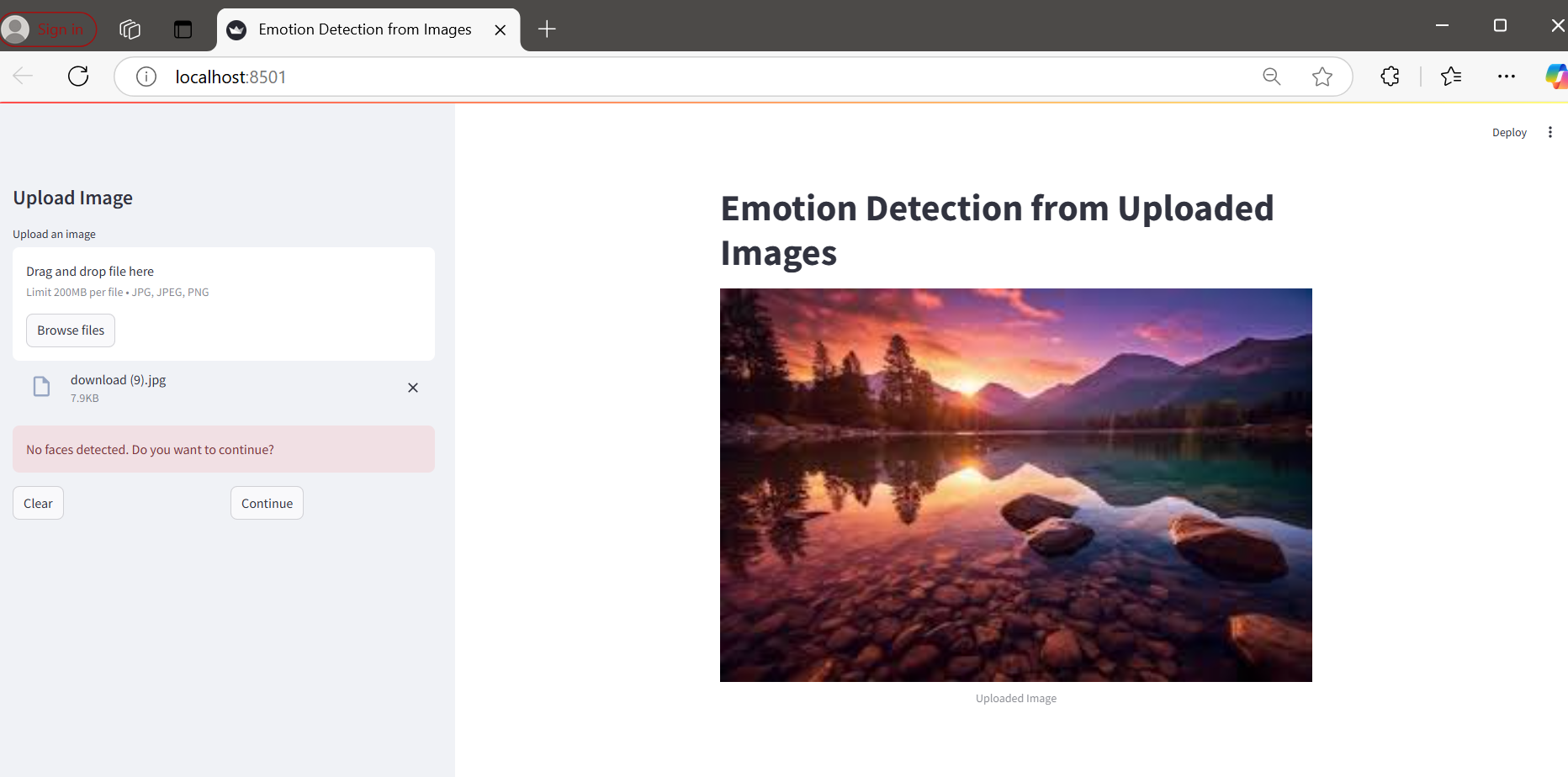


Case 2b: User chooses to continue with the image processing for detecting the emotion

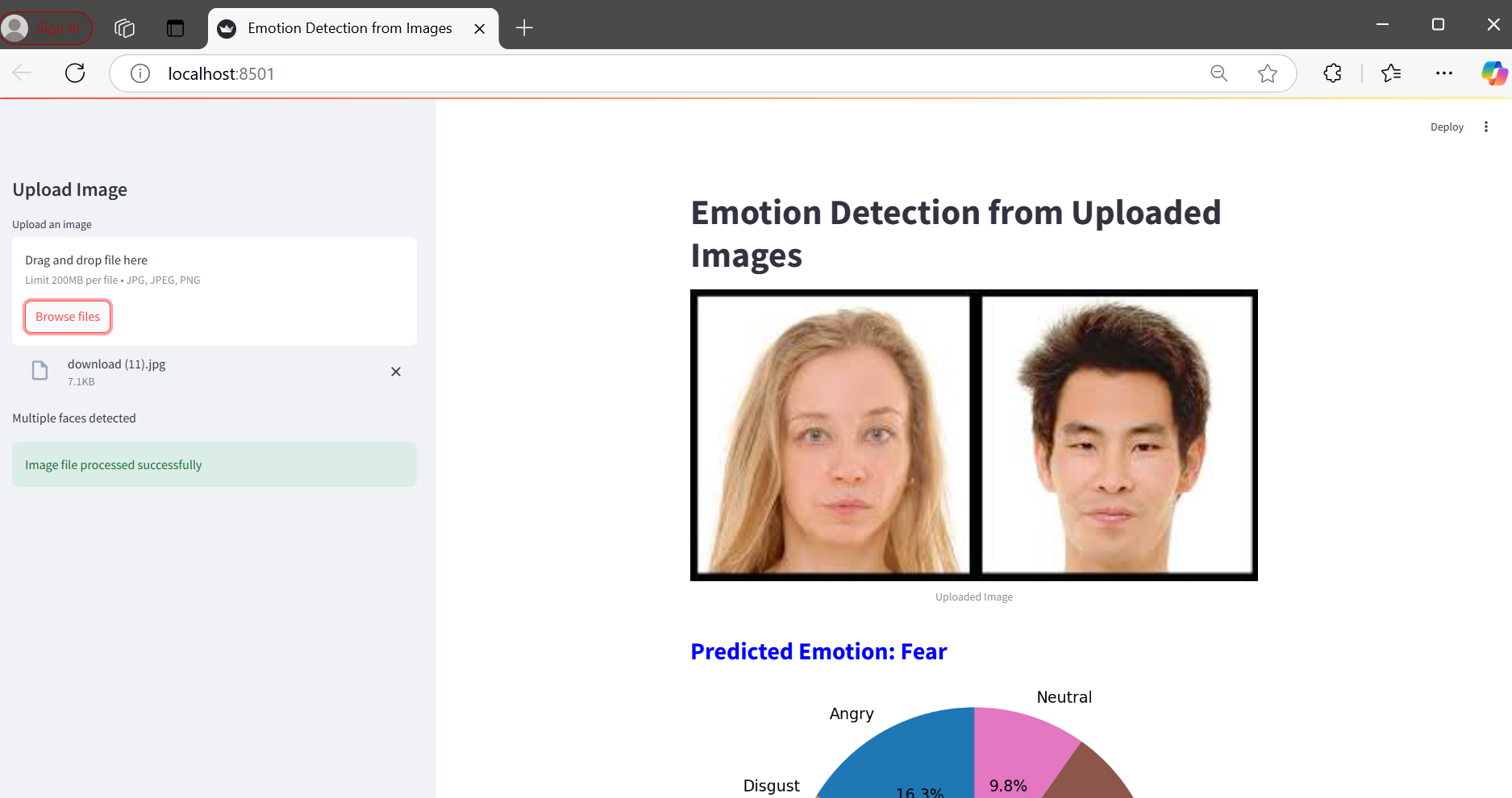


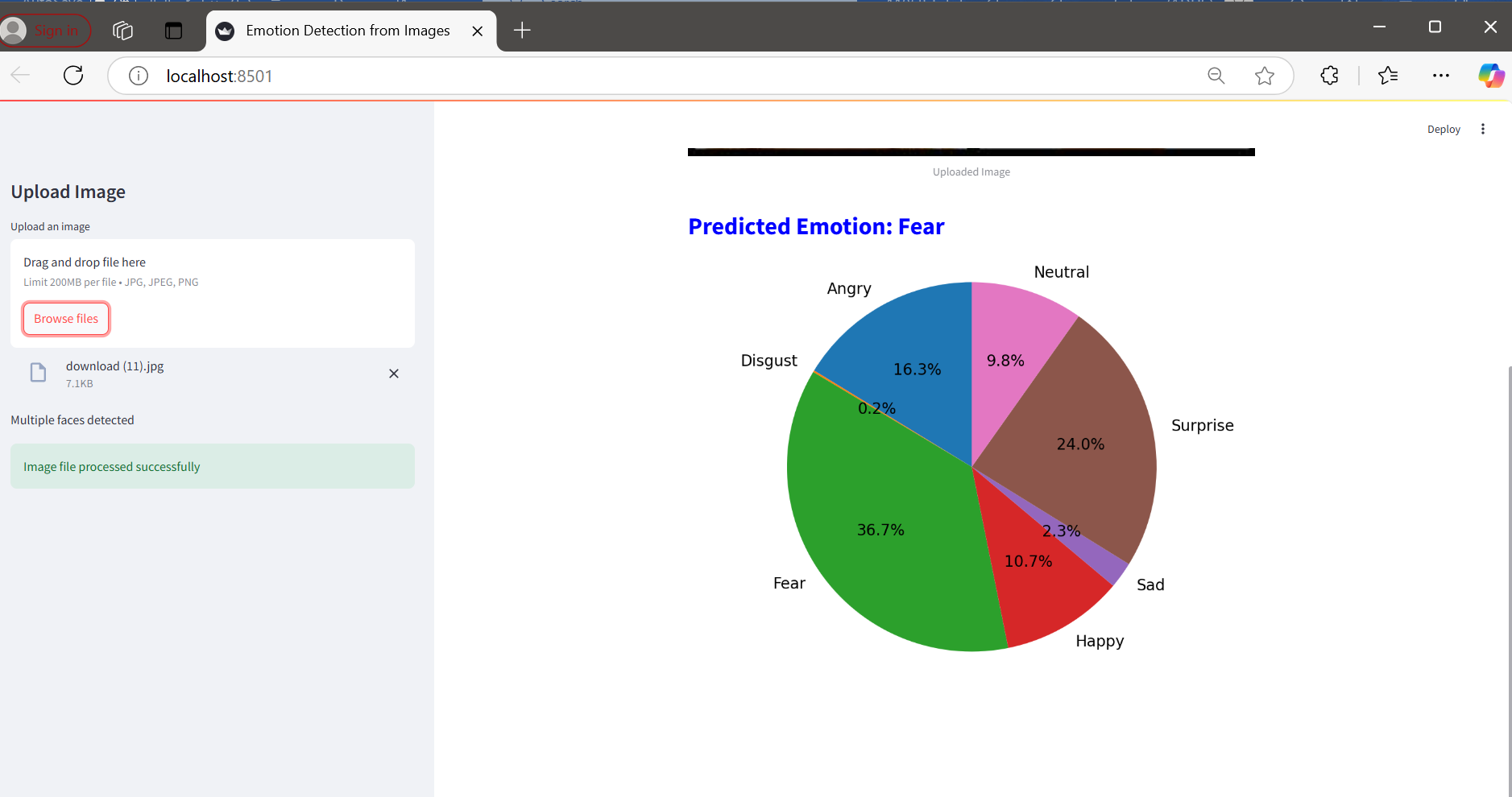


Case 2 c:Another no face detected

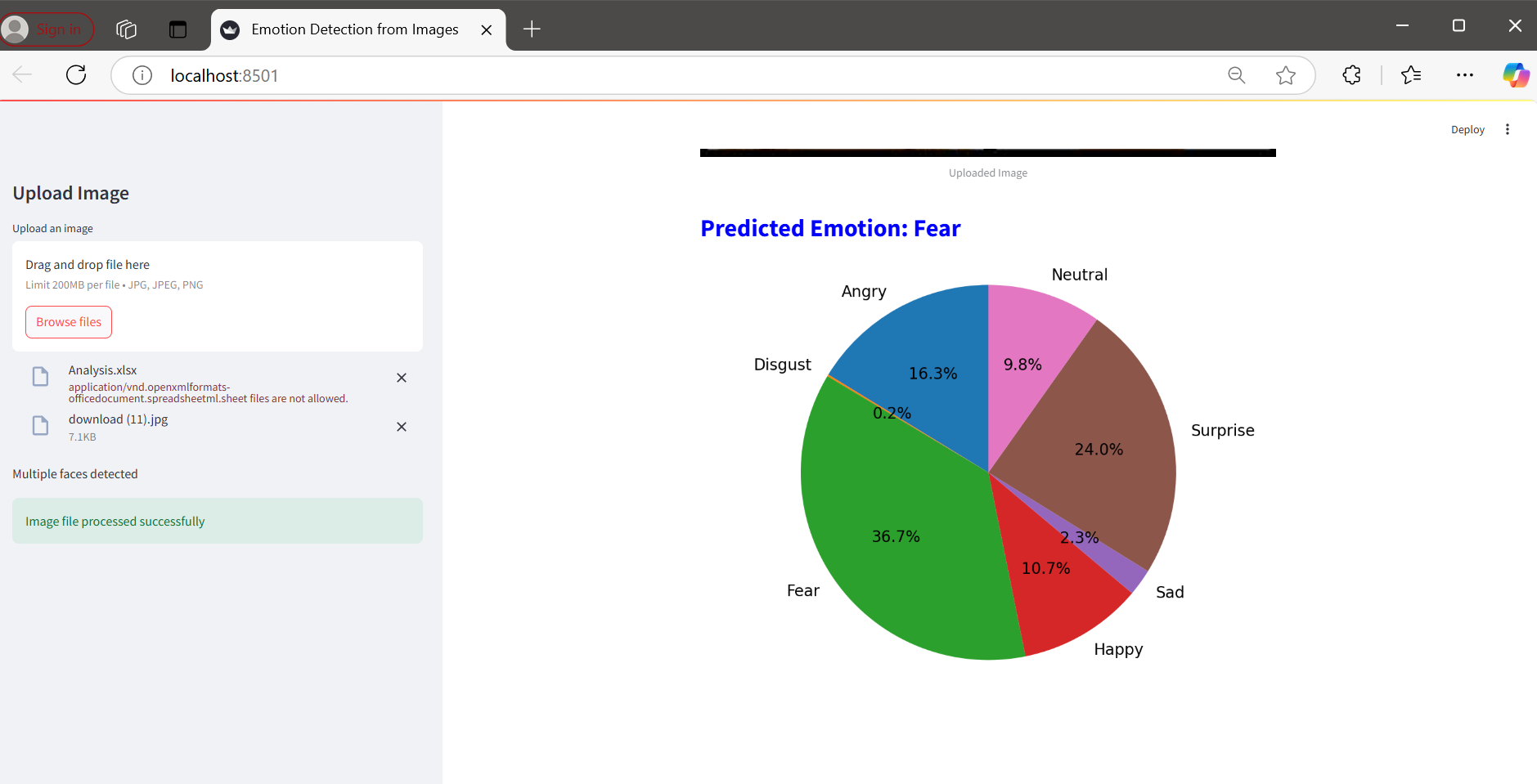


Case3: Multiple faces detected





Case 4: Trying to upload files other than .jpeg,.jpg,.png



**ETHICAL CONSIDERATIONS FOR EMOTION DETECTION TECHNOLOGY:**

1. **Privacy Concerns**
   * **Informed Consent**: Users should be fully aware of how their data is being used, including facial images and emotions.
   * **Data Security**: Ensure sensitive data (e.g., images) is stored securely and processed according to privacy laws.
   * **Data Retention**: Data should not be stored longer than necessary, and users should have the right to request deletion.
2. **Bias and Fairness**
   * **Algorithmic Bias**: Emotion detection models should be trained on diverse datasets to avoid misclassifying emotions for certain groups (e.g., based on race, gender, age).
   * **Cultural and Gender Bias**: Models must account for differences in emotional expression across cultures and genders.
   * **Bias Mitigation**: Use fairness-aware algorithms and conduct regular audits to ensure fairness.
3. **Accuracy and Impact of Misclassification**
   * **Consequences of Misclassification**: Misinterpreting emotions can have harmful effects in sensitive sectors like healthcare, law enforcement, and customer service.
   * **Over-reliance on Technology**: Emotion detection should be a supplementary tool, not the sole decision-maker.
4. **Ethical Use in Surveillance**
   * **Involuntary Monitoring**: Emotion detection should not be used for constant surveillance without consent, especially in public spaces or workplaces.
   * **Workplace Monitoring**: Use with caution in monitoring employees to avoid invasion of privacy or emotional manipulation.
5. **Potential for Manipulation and Exploitation**
   * **Emotional Manipulation**: Avoid using emotion detection for exploitative marketing or psychological manipulation.
   * **Psychological Impact**: Constant monitoring could cause anxiety and affect individuals' self-image.
6. **Transparency and Explainability**
   * **Clear Communication**: Users should be informed about how their emotions are analyzed and classified.
   * **Model Explainability**: Emotion detection systems must be interpretable so that users can understand why decisions are made.
7. **Regulation and Ethical Guidelines**
   * **Ethical Use**: Implement guidelines to ensure the responsible use of emotion detection, with clear regulations for sensitive sectors.
   * **Accountability**: Developers must ensure that their systems are transparent, fair, and accountable to prevent misuse.

### CONCLUSION:

Emotion detection is an emerging technology with high **business potential** across industries. In **healthcare**, it can drive better mental health care; in **education**, it fosters personalized learning; in **customer service**, it enhances user experience; in **market research**, it enables targeted strategies; and in **HR**, it can promote employee well-being. By leveraging emotion detection, businesses can improve both **operational efficiency** and **customer satisfaction**, offering a competitive edge in their respective sectors.