```python
import cv2
from tensorflow.keras.models import load_model
from tensorflow.keras.utils import get_custom_objects
import numpy as np

# Load the pre-trained emotion detection model
get_custom_objects().update({'fbeta': lambda *args, **kwargs: None})
model = load_model('emotion_model.h5', compile=False)

# Emotion labels dictionary
emotion_dict = {
    0: "Angry 😠",
    1: "Disgust 🤢",
    2: "Fear 😨",
    3: "Happy 😊",
    4: "Sad 😢",
    5: "Surprise 😲",
    6: "Neutral 😐"
}

# Start webcam capture
cap = cv2.VideoCapture(0)

# Load Haar Cascade face detector from OpenCV
face_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_frontalface_default.xml')

while True:
    # Read each frame from webcam
    ret, frame = cap.read()
    if not ret:
        break

    # Convert to grayscale
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    # Detect faces in the frame
    faces = face_cascade.detectMultiScale(gray, 1.3, 5)
```

```
38
39      for (x, y, w, h) in faces:
40          roi_gray = gray[y:y+h, x:x+w]
41          cropped_img = cv2.resize(roi_gray, (48, 48))
42          img = cropped_img.astype('float32') / 255.0
43          img = np.reshape(img, (1, 48, 48, 1))
44
45          # Predict emotion
46          prediction = model.predict(img)
47          emotion_index = int(np.argmax(prediction))
48          emotion = emotion_dict[emotion_index]
49
50          # Show what the model is seeing
51          cv2.imshow("Cropped Face", cropped_img)
52
53          # Print prediction confidence
54          print("Prediction array:", prediction)
55          print("Predicted emotion:", emotion)
56
57          # Draw rectangle and emotion text
58          cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)
59          cv2.putText(frame, emotion, (x, y-10),
60                      cv2.FONT_HERSHEY_SIMPLEX, 0.9, (255, 255, 255), 2)
61
62      # Display the video with emotion labels
63      cv2.imshow("Real-Time Emotion Detector", frame)
64
65      # Press 'q' to quit
66      if cv2.waitKey(1) & 0xFF == ord('q'):
67          break
68
69  # Release webcam and close window
70  cap.release()
71  cv2.destroyAllWindows()
72
```