Project Spec: Egg Quality Detection

Name: Lakshmi Jitta
Course: HCI 584
Project Name: Egg Quality Detection

1. General Description of the Project:

Egg grading is an essential task in poultry production and food quality control, but manual assessment of egg quality based on shell color, cracks, and size is time-consuming, inconsistent, and difficult to scale. This project aims to develop a prototype system that automates these assessments using basic computer vision techniques. The first version of this system will focus on identifying a single egg in an image and classifying its quality based on color, shape, and visible cracks.

The primary goal is to use Python and OpenCV to implement the core logic that enables automatic detection and classification of a single egg in a still image. The system will analyze the egg to determine whether it is a "Good" or "Bad" egg based on color clustering in the HSV color space. It will also estimate egg size using a reference marker of known dimensions and flag structural irregularities by detecting irregular edges that may indicate cracks.

The initial version will run as a command-line tool, where the user can input an image file. The system will output a summary of results to the terminal and save an annotated image showing the egg with its classification and size. For future versions, the project will include a graphical user interface (GUI) built with **Tkinter**, which will allow users to select input images, view results in a table format, and adjust classification parameters. Later versions may also support multiple eggs per image.

Though not included in version 1, future extensions may be explored using pre-trained models from OpenCV or integrating advanced vision tools.

2. Task Vignettes

Vignette 1: Load Image and Start Detection – We open the command line and run the Python script, providing the path to an image showing a single egg placed on a flat surface with a ruler as a reference. The script loads the image and displays a console message indicating that detection has started.

- Technical Notes:
    - Use cv2.imread() to load image
    - Detect egg using cv2.findContours() on blurred and thresholded image
    - Preprocessing includes Gaussian blur and grayscale conversion

Vignette 2: Analyze Egg Shape and Color. The script processes the contour and isolates the egg region. It converts the region to HSV and uses KMeans clustering (1 cluster) to find the dominant color. Based on HSV thresholds tuned from sample data, the egg is labeled as "Good or Bad."

- Technical Notes:
    - Convert region using cv2.cvtColor() to HSV
    - Use KMeans(n_clusters=1) from sklearn.cluster
    - If contour is small or irregular, discard as noise
    - Color rules based on thresholds from manually labeled egg data

Vignette 3: Estimate Egg Size with Marker The system uses a visible marker (e.g., a printed 5 cm square or a ruler) to calculate a pixel-to-centimeter ratio. The egg's bounding box is measured and converted into real-world units.

- *Technical Notes:*
    - Detect marker based on known color or shape
    - Calculate size using cv2.boundingRect() and known marker size
    - Store estimated egg dimensions

Vignette 4: View and Save Results. The processed image is saved with annotations showing the contour, classification, and size. A printed summary includes the egg's quality label and size.

- Technical Notes:
    - Use cv2.putText() to draw labels on image
    - Save image using cv2.imwrite()
    - Print formatted summary in terminal (egg class, size)

3. Technical Flow

Image Input	• Load image from file path (JPG or PNG) • Optional: Support webcam snapshot (future)
Preprocessing	• Resize, convert to grayscale • Apply Gaussian blur • Apply adaptive thresholding or Canny edge detection
Egg Detection	• Use cv2.findContours() to extract egg shape • Filter based on size/shape (area, aspect ratio)
Color Classification	• Convert egg region to HSV • Apply KMeans to estimate dominant color • Classify as Good/Bad based on rules from known samples
Size Estimation	• Locate marker, compute pixel:cm ratio • Measure bounding box for the egg
Crack Detection (Basic)	• Compute edge density or contour jaggedness • Flag highly irregular eggs
Output	• Save annotated image • Print summary to console
Data Types:	• Input: Image files (JPG, PNG) • Intermediate: Arrays, HSV tuples, float values • Output: Annotated image, text summary, optional JSON/CSV (future)

4. Final Self-Assessment

Biggest Changes from Sketch: Originally, the project focused only on detecting color and cracks. After instructor feedback, I added the size estimation feature using a visual marker and committed to developing a GUI in version 2. I also clarified that version 1 will work with single-egg images to simplify detection and improve reliability. Multi-egg support may be considered in later stages.

Confidence Level: I am moderately confident in my ability to complete version 1. The image processing parts using OpenCV and KMeans are already partially tested. The hardest part will be integrating size estimation accurately from the marker.

Biggest Risk: The size estimation depends heavily on the visibility and clarity of the physical marker. If it's not detected reliably, the size computation will fail. I may need to experiment with marker shapes and contrast.

Help Needed: I might need support designing the GUI in version 2 and refining the color thresholds for classification under different lighting conditions. I also plan to ask for feedback after I finish version 1 to make sure the GUI additions are meaningful and robust.