

Planter Bot

Task 1B – Overlay flowers on detected color objects

Goal:

- To learn Python and use it to:
 - Identify colours of objects
 - Identify the position of objects in the frame
 - Overlay another image with a transparent background on the objects one at a time

Please find the *Task-1B_main.py* file in the nested folder named “2. Task_Description\Video”. Modify the sections of *Task-1B_main.py* marked for the same, to accomplish the following:

Given:

- A 7-second video having frame rate of 30fps containing objects which appear over time as the video progresses. The resolution of each frame is: 1280x720. Find this video in the nested folder named “2. Task_Description\Video”
- Each object in the video is defined by three features, viz. **Color, Shape** and **Position** in the frame.
- Each object will be of one of three primary colors, viz. **Red, Green and Blue**.
- Each object will be of one of the six shapes, viz. **Triangle, Rhombus, Trapezium, Pentagon, Hexagon and Circle**.
- The still images containing the above mentioned object shapes have been provided in the “Sample Shapes” folder. You can use these shapes as templates for making your own frames and videos subsequently **for practice**.
- The flower images for overlay are provided in a subfolder named “Overlay_Images” within the nested folder “2. Task_Description\Video\”. The flowers to be overlaid for each of the primary colors is given in Table 1 below; where the first column is the primary color and second column is the name of the image in the said folder.

Table 1: Overlay images corresponding to the primary colors

Primary Color	Image Name (Overlay Image)
Red	yellow_flower.png
Green	red_flower.png
Blue	pink_flower.png

For example, yellow_flower.png contains the image of a sunflower as shown in Figure 2.



Figure 2: yellow_flower.png

Problem Description:

Split the Video into individual images or frames.

For a given frame/individual image find:

1. The color
2. The shape, and
3. Position -- centroid of each object-- in the *frame* where the flower image has to be overlaid.

Your code should return a 7 second video with the following features:

1. The video should be displayed at 30fps having the same aspect ratio and resolution as the original video viz. 1280x720
2. Each colored object appearing on the video must be overlaid with the corresponding flower image (Refer to Table 1)
3. As soon as each new object appears the overlay for the previous object should disappear .i.e. **at a time only the latest occurring object will have an overlay.**

For each color in the image, overlay its corresponding flower image which is a png with a transparent alpha channel (BGRA OpenCV format). An example is shown in Figure 3. Note that with reference to Table 1, the yellow flower is meant to be overlaid on a red colored object.



Figure 3: Yellow Flower overlaid on Red Rhombus

You can practice overlay mechanism on still images before performing the same task on a video input. The “**blend_transparent**” method in the Task-1B_main.py Python script file is provided for the purpose of overlay. Do not modify this code segment. You are expected to understand it and call it in such a way that overlay is successful.

Required Output:

1. The Python script file named Task-1B_main.py with the function “**main**” populated with your logic of solving the task.
2. A csv file with a List containing the properties of the objects viz. **Color-Shape-CentroidX-CentroidY** of the objects as they appear in the video in the sequence in which the object appears.
3. An output video with overlay process happening as the colored objects appear in the video.

To do:

1. Open **Task-1B_main.py** located in the folder named **“Video”**. It has three functions:

- **main():** It returns the required List for the **writcsv()** function. It should also write the output video to file. This is the method where you write your logic to solve the task.
- **writcsv():** This function expects four parameters - **Color, Shape, CentroidX and CentroidY** as arguments and writes these properties to file, one object at a time into a **“result1B_<TeamID>.csv”** file. For the red rhombus in the frame as shown in Figure 3 above, the properties are to be written in the format: [**“Red-Rhombus-191-361”**].
Do not edit this function. However, you can edit the global variable - “filename” with your TeamId in the Task-1B_main.py Python code file. Please use your eYRC team id while actually naming the file. Please also name the output video with respect to its corresponding input as - **“videooutput.mp4”** for an input video named **“video.mp4”**.
- **blend_transparent():** This function has been obtained from a general Google search of **“how to overlay a transparent image on another image”**. It has been slightly modified to perform the overlay task for our problem statement. **Please do not modify this function.** You are expected to understand the functionality of this function and call it from main (within your code) to perform the overlay accurately.

IMPORTANT: Do not change names of any of these functions.

Rules:

1. You need to write a **generic program**. Note that we have provided one 7 second long video with multiple objects of different colors. In addition, ***your code will be tested on several undisclosed “videos” when you submit your code.***
2. Use basic knowledge of Video processing using OpenCV to convert video to a series of frame or individual images.
3. Objects are not rotated or differently oriented with respect to each other.
4. In case objects are found in a frame, please overlay the flower image specific to that color.

Happy Learning!

All The Best!!!