**Slack DevIncept**

**Open CV Course**

**Module 2**

**1) Segmentation**

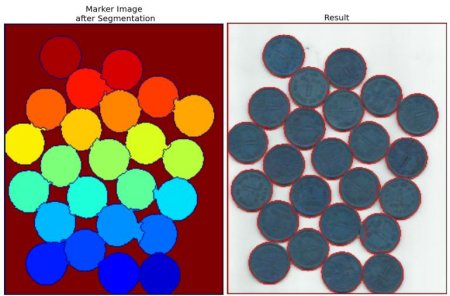
We implement marker-based image segmentation using watershed algorithm. Any grayscale image can be viewed as a topographic surface where high intensity denotes peaks and hills while low intensity denotes valleys. You start filling every isolated valley (local minima) with different coloured water (labels).

OpenCV implements a marker-based watershed algorithm where you specify which are all valley points are to be merged and which are not. It is an interactive image segmentation. What we do is to give different labels for our object we know. Label the region which we are sure of being the foreground or object with one colour (or intensity), label the region which we are sure of being background or non-object with another colour and finally the region which we are not sure of anything, label it with 0. That is our marker. Then apply watershed algorithm. Then our marker will be updated with the labels we gave, and the boundaries of objects will have a value of -1.

Input Image:



Output Image:



**2) Contours**

Contour can be defined as an outline or a border of any object. Contours in Python is a list of contours of an object, which can be an image, a frame from a video and so on. Each Contour in the list Contours is a point represented as (x, y) which represents the position of the point with respect to the two axes. The contoural points usually have the same intensity.

Contours are mainly used in Object Detection.

Determining contours involves multiple steps as the image needs to get ready for the process. After reading and storing the image in a variable, the image needs to be converted to either hsv or grayscale to ease the processing.

To obtain all the contoural points on the image, drawing the contours is necessary. To do this the OpenCV has the drawContours() function that draws or plots all the contoural points onto the image.

To make contouring more efficient, the cv2.Canny() function comes handy to ease the process for users. This function makes the edge detection job done for the users so that the contouring becomes easier.

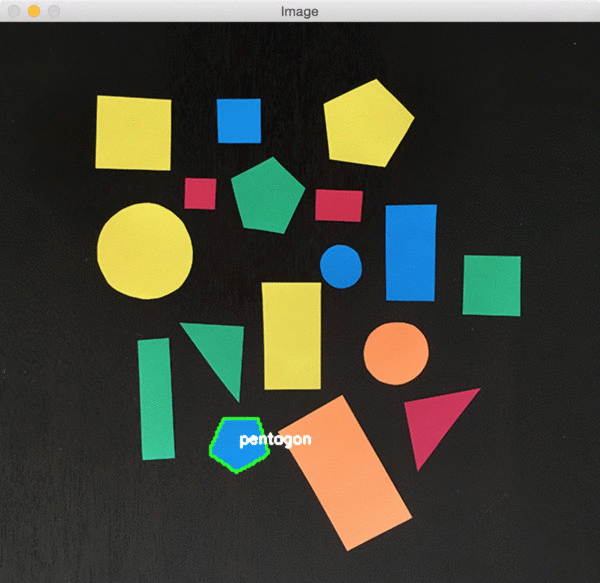
Video is undoubtedly a series of image frames. This concept makes video processing easier, especially when using OpenCV library. Contouring a video is nothing but contouring the individual frames of the video. So, video contouring is nothing but contouring multiple images.

**3) Shape Detection**

Shape Detection is nothing but finding different shapes in an image and identifying them. Using OpenCV, computers can be made able to detect shapes in an image by feeding them with some information and applying certain conditions and allow them to act based on the decision. This process involves multi-steps:

* Installing pip and imutils packages.
* Importing the necessary libraries.
* Creating the ShapeDetector class.
* Loading and processing the image and finding the contours.
* Finding the centre of each contour and detecting the shape.
* Displaying the output image.

Input Image:

Output Image:

**4) Feature Detection and Feature Description**

If we see an image or object, we closely look into in and try to find a pattern to distinguish that image or object from another and it all happens in the background.



In the above image when we look at it our brain starts to find a pattern, like the black box is sky and the white box is the windows of the building and the red box shows the building walls. These all are the features that we find just by looking at the image. Once we find this feature in this image we can find them in other images or objects and this process is called Feature Detection. After finding a feature we describe the feature and this is called Feature Description. Same is done by computer to describe the region around the box so that it can find that in other images and objects.

**5) Face and Eye Detection**

Face and Eye detection works on the algorithm called Haar Classifier.

Haar Classifier is a machine learning based approach where a function is trained from a lot of positive and negative images i.e. with face and without face.

Initially the algorithm needs lots of positive images (with face) and negative images (without face) to train the classifier (algorithm that sorts data in categories of information). Once all the features and details are extracted, they are stored in a file and if we get any new input image, check the feature from the file, apply it on the input image and if it passes all the stages then the face is detected. So this can be done using Haar Features.

So in short, Haar Classifier is a classifier which is used to detect the object for which it has been trained for from the source.

**6) Flask**

Flask is a web application framework written in Python. A web framework is a collection of modules and packages which facilitates or supports the user to create dynamic web applications. It facilitates creation, development and publishing of web applications. Flask is based on the Werkzeug WSGI toolkit and Jinja2 template engine. It includes a built-in development server, unit testing support, and is fully Unicode-enabled with RESTful request dispatching and WSGI compliance.

WSGI (Web Server Gateway Interface)

Web Server Gateway Interface is a simple calling convention for web servers to forward requests to web applications or frameworks written in the Python programming language. It acts as an interface between web servers and web applications.

Jinja2

Jinja2 is a template engine written in pure Python. It combines a template with a certain data source to render dynamic web pages.

Flask works by first creating a web page with flask and connecting it to HTML. Anytime any user sends information on net or goes to the search bar, the HTML connects the user. The flask framework looks for HTML files (templates) in a folder called Templates. Before sending the template over, Python code is executed which injects variables, code etc.

Report made by:

**Jigyasa Bajpai**

Email ID – [jigyasa10feb@gmail.com](mailto:jigyasa10feb@gmail.com)

Phone number – 9930309651