Inlämningsuppgift#1 – individual

Publishing: 2025-12-15

Due date: 2026-01-10 kl 23:59

Betyg : G >70% , VG > 90%

Project 1:

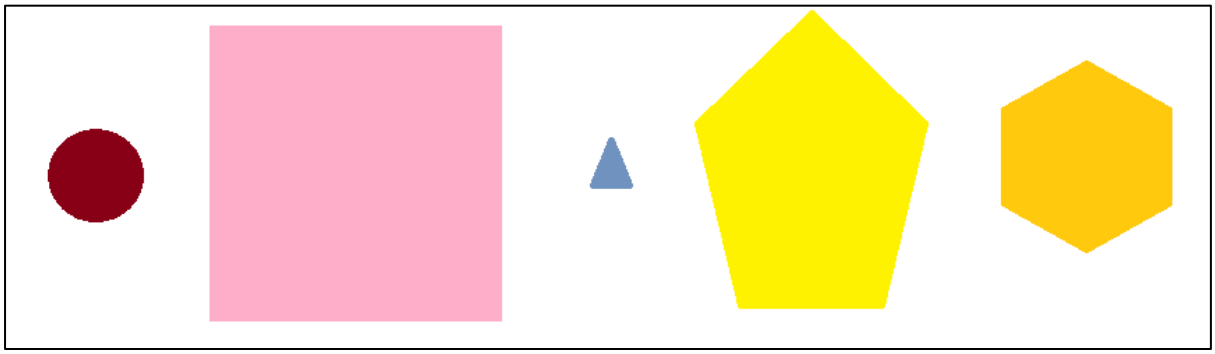
Sorting objects by area is a commonly used technic into electronics industries. It is based on identification of the object per **regular** shapes then sorting them as preparations to next step which is find the best possible design on the silicon chipset.

Your task is to create python program that take an image contains different shapes and:

> generate descending sorted list per area of shapes found and related name. ***(25 points)***

Example that you can use for demonstrating:

Input image is:



the list will be:

sort	Shape name	Area _ numbers just for example. In pix
1	square	100
2	pentagon	92
3	hexagon	83
4	circle	50
5	triangle	45

Project 2: **Image classifications:**

1- We need to create our class called image_classification that take in an **image**, then detect objects into this image and grab each image detected with confidence more than 90% and add them (save them with image_detected_names_num.jpg) into these main categories(folders): ***(25 points)***

Cat#	Category	Image detected
1	Human	person
2	Vehicles	bicycle car motorbike aeroplane bus train truck boat
3	Animal	bird cat

		dog horse sheep cow elephant bear zebra giraffe
4	Sport and Lifestyle	backpack umbrella handbag tie suitcase frisbee skis snowboard sports ball kite baseball bat baseball glove skateboard surfboard tennis racket
5	Kitchen stuff	bottle wine glass cup fork knife spoon bowl
6	food	banana

		apple sandwich orange broccoli carrot hot dog pizza donut cake
7	In house things	chair sofa pottedplant bed diningtable toilet tvmonitor laptop mouse remote keyboard cell phone microwave oven toaster sink refrigerator book clock vase scissors teddy bear hair drier

		toothbrush
8	MISC	For all other things we don't find into this column 😊

2- Once done with image as input, we need to upgrade it to **video** as input. ***(10 points)***

Project 3 : MyImageProcessor

1- Using Numpy, OpenCV and matplotlib Let's create a class called MyImageProcessor that has:
   a. Constructor (__init__) that read in the path or image file name. ***(5 points)***
   b. Create method called bgr_2_rgb_convertor(self) this will take the image path that we already read into constructor and show it in real RGB colors AND return it as Numpy array. ***(5 points)***
   c. Create method called bgr_2_gray_scale_convertor(self) this will take the image path that we already read into constructor and show it in gray scale colors AND return it as Numpy array. ***(5 points)***
   d. Create method called _50_percent_resizer(self) this will take the image path that we already read into constructor and show it in real RGB colors and 50% of the width and height of original image AND return it as Numpy array. ***(5 points)***
   e. Create method called image_writer(self,output_image_path_and name) that takes string of the output image name with or without path, and take the image that we already read into constructor and save it in RGB coloring. ***(5 points)***

f. Create method called frame_it(self,output_image_with_frame_path) that draws a **RED** frame(rectangle) around the image with 20 pixel thickness, And save it in RGB according to path into string argument output_image_with_frame_path, and return it as Numpy Array. ***(5 points)***
g. Create method called find_center(self,output_image_with_center) that draws **BLUE** point in **center** of the image we already read in constructor and write in good text size , “image center” And save it in RGB according to path into string argument output_image_with_frame_path, and return it as Numpy Array. ***(5 points)***
h. Using CascadeClassifier create method detect_faces(self) that **return** the image with **RED** rectangles around faces and faces_counter variable that has how many faces found in the image. ***(10 points)***