

## **CS445 : Final Project**

### **Image classification and similarity detection**

#### **Team Members :**

Senthil Ramachandran (skr6)  
Rujitha Vennugopalan (rujitha2)  
Nandhini Gunalan (gunalan3)

#### **Motivation :**

The aim of this project is to detect and extract the object in an image, classify the object into one of the existing object classes and detect similarity with the list of sample images. An object's spatial, textural, and contextual properties are all being considered in classification and similarity detection.

#### **Resource requirements:**

The technical requirements for this project are as follows:

ipywebRTC  
numpy  
Pandas  
Tensor flow  
sklearn  
Deep Learning and Algorithm (R-CNN)  
pytorch  
tqdm  
matplotlib  
Jupyternotebook

## Object Detection and Extraction

This image used to detect and extract objects using contouring.



The input image is first converted into a gray image.  
cv2.getStructuringElement and cv2.dilate were used to set threshold.  
cv2.boundingRect is used to create contours.

Input file : Object\_Extraction/input/desk1.jpg  
Contoured image : Object\_Extraction/output/desk1\_contoured.jpg  
Extracted objects: Object\_Extraction/output/desk1/

## Object Classification

For object classification the existing training data is created by adding noise to the images followed by creating a training model. Then new unseen images with the objects are tested. The object is then placed in one of the existing predicted classes.

**Test image:**



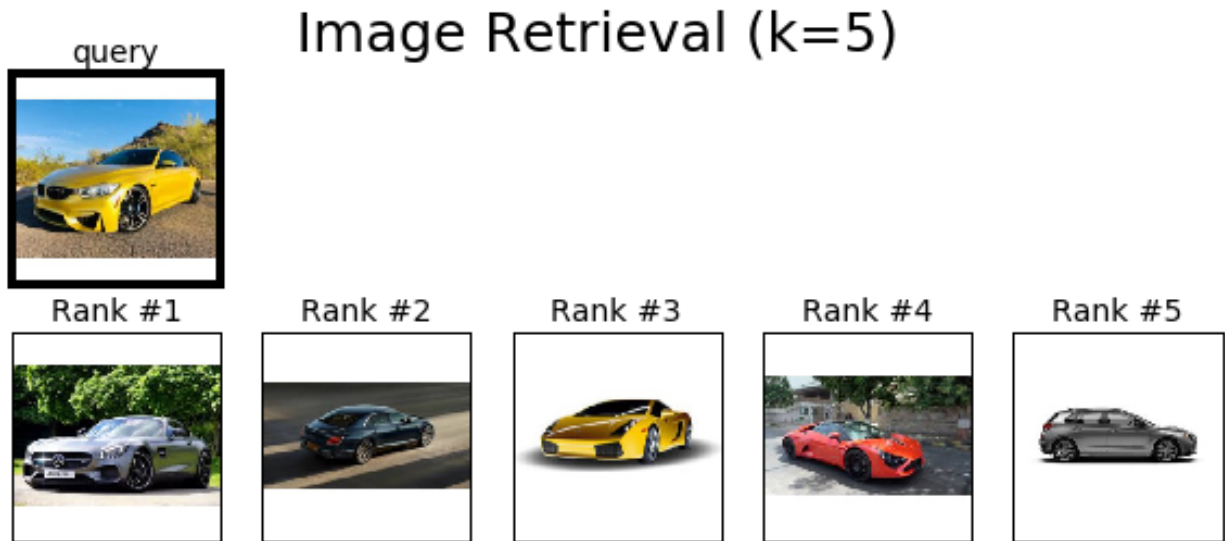
**Output:**

Output class : Car  
Percentage probability : 98.9%

car : 98.90155792236328

## Similarity detection

For Similarity detection sklearn nearest neighbors is used. A train and test set of data was chosen and the has three pre labeled classes car, ball and book. This example was generated using k=5 nearest neighbors.



### Conclusion:

The project is developed as a one stop solution to object detection, classification and similarity detection. This can be used in various areas that require objects to be identified and classified even when the image quality is poor. This project would also help in fields that require huge number of images to be processed.

### References:

<https://medium.com/object-extraction-from-images-using-opencv-python>

<https://www.pyimagesearch.com/2014/10/20/finding-shapes-images-using-python-opencv/>

<https://www.pyimagesearch.com/2017/12/11/image-classification-with-keras-and-deep-learning/>

<https://github.com/tzutalin/labelImg> - Used for labeling input images