**Name of the Guide:** Dr. Sonali Patil

**Group members:** Rohit Sharma - Roll no.: 1713054

Rhythm Shah - Roll no.: 1815058

Krutika Bhatt - Roll no.: 1911069

**Sharp Corner Detector**

**Introduction:**

This project revolves around creating an application to detect and mark sharp corners of objects in the given image and determine the movability score of each object present.

In an image, corners are the region with large variation in the intensity of pixels in all directions. The sharp corner detection methods uses Harris corner detection and Shi -Tomasi corner detection. Points are further filtered by edge detection in the region of interest.

Movability or capability of object to move is a concern of area. Larger the area covered by object, less the movability score. In an image, the movability score is determined by comparing the areas of the objects with one another and then assigning them the score corresponding to their areas.

**Implementation Details:**

We will make the implementation of code easy by breaking it into three parts.

1. Corner Detection for Single object images:

Step 1: The image is converted to gray-scale image and binarization is performed using

cv2.THRESH\_BINARY\_INV + cv2.THRESH\_OTSU

In OTSU binarization, we avoid choosing a value and let it decide the value automatically according to an image. This improves the noise filtering.

Step 2: We use Shi-Tomasi method for this detection i.e. goodFeaturesToTrack() method. We filter the points obtained by taking the largest contour area. So by this method we get points only of largest contour.

Step 3: The points which are common in contour region and points returned by goodFeaturesToTrack() are highlighted. This is how we detect corners of a single object.

1. Corner Detection for Multiple objects images:

Step 1: Convert an image into gray-scale image. We extract the background of image by simply doing a floodfill operation from pixel(0,0).

Step 2: We use Harris corner detection method to get corner points of objects. But we even get those points which have high intensity color change in pixels. So to avoid them we use convex Hull method. This evades all the points which are not the corners.

Step 3: The corners which are being detected are highlighted using drawing circle method. This is how we get all the points which are corners of the objects.

1. Movability Score of objects:

Step 1: We convert the image into gray-scale image. Then we apply blurring effect on it so that we can use canny detection method. We dilate the image so we get only the boundaries of the objects.

Step 2: We then find different contours using findContours() method. Now the contours detected surely have some area. So we find areas of each contour and it is stored in the form of list.

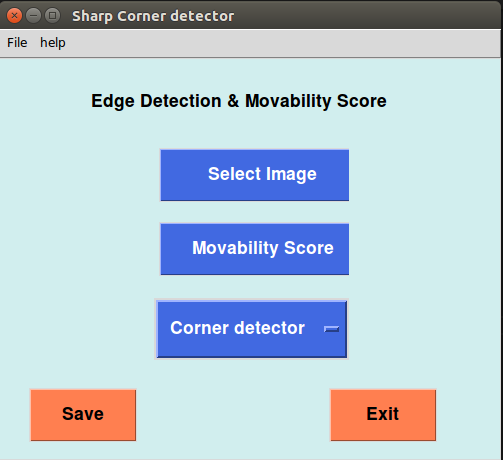
Step 3: Now we divide the list of areas into three equal parts. The areas associated with the first part have the movability score as high because they have smaller areas so they are easily movable. Second part has been assigned movability score of medium because they are not that easy to move and have slightly larger areas. Third part have contours with larger areas. So they are assigned low movability score because they are difficult to move.

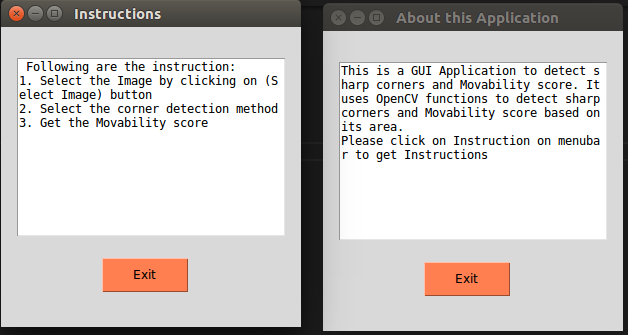
This is how we assign movability score to each object.

**Output:**

A GUI application is made with the help of tkinter and OpenCV that comprises of all the functions used to detect sharp corners and get the movability score.

1. The GUI application is made user-friendly by adding menubar at the top of the frame. User can click on help to get information and instruction to use this app. Further functions to open a new image, save and exit is made available in File on menubar

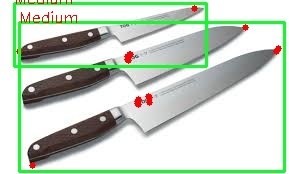
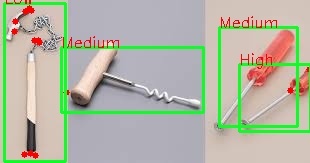




1. The project gives fine result on many images. Following are the output Images after Sharp corner detection and Movability score.
2. Single Objects – Shi-Tomasi Method

1. Multiple Objects – Harris Corner Detection Method



Let’s discuss about Shi-Tomasi and Harris corner detection outputs.

For Single object images:

We prefer Shi-Tomasi method because it detects the corner points accurately while when we use Harris corner detection we get many unnecessary points too. We can observe this by looking at the outputs obtained.



Harris corner detection Shi-Tomasi Method

For multiple objects images:

In case of multi-object images, Shi-Tomasi gives good clearance and distinct points but misses some sharp corners due to filtering and noise removal functions performed on the image. Thus in this case, harris corner detection is used so that the point of interest is not lost by getting large dataset of points which are filtered further. Here is the following example of use of Harris detection and Shi-Tomasi methods on the raw Image with light-reflection, shadow and multiple places with drastic Intensity change.

Harris corner detection Shi-Tomasi method

But as we can see there are many unnecessary points marked in Harris detection , we need to apply filtering techniques to remove unnecessary points. But in Shi-Tomasi method, there are no sharp corners detected. So there is no point in using this method for multiple objects.

After filtering we can see the output of Harris Corner detection as follows:



So here we see how Shi-Tomasi method is useful for single object images and Harris corner detection for multiple objects images.

**Conclusion:**

The sharp corners and Movability score of an object in a given image is detected successfully. The GUI application makes the project user-friendly and attractive. Concepts related to Sharp corners or places of intensity change in an image is studied and implemented. We also conclude that Shi-Tomasi method gives best results for a single object image and Harris corner detector gives the best result for multiple objects images. And also the movability score based on relative area of objects itself is tested and successfully implemented. Further work and research can be achieved on the same topic with different complex techniques to get more refined and better results which would be industrially useful.