**3. IMAGE PROCESSING – SKYSTONE DETECTION**

**3.1. IDEA OF THE TASK:**

* Capture the Front ground.
* Use Colour detection mechanism and Contour Area property.
* Detect the cascaded stones.
* Tracking the skystone in the mentioned directions.
* Optimise the Time Taken for the task.

**3.2. OVERALL DESIGN:**

**Processing steps we made:**

* Colour Detection mechanism for Yellow colour Detection.
* Contour Area property to extract the cascaded stones.
* Optimised the implementation to run the Autonomous period fast.

**19 Seconds**

**Identify**

**Contour**

**Area**

**Colour Detection**

**Left / Centre / Right**

**Cascade of Stones**

**Yellow**

**Objects**

**Captured Image**

**Optimized To:**

**4.5 Seconds**

**Identify**

**Contour**

**Area**

**Colour Detection**

**Left / Centre / Right**

**Cascade of Stones**

**Yellow**

**Objects**

**Captured Image**

**3.3. COLOUR DETECTION:**

We used colour detection mechanism to detect the yellow coloured objects for the captured image when autonomous period begins and obtained the binary image as output.

**Software Libraries:**

1. cv2
2. numpy

We used averaging **blur()** function in colour detection to detect the perfect yellow colour using the Binary to HSV conversions of applied frequency for yellow colour.

**Processing Steps:**

* Bot captures the front image.
* The captured image made blurred using Averaging Blur().
* The blurred image then converted to HSV.
* Using Numpy array, the frequencies of the yellow colour is entered.
* Using the above conditions, the output image will be displayed.

**Implementation:**

**//Importing libraries**

**import cv2**

**import numpy as np**

**//Capturing the image**

**\_,img = cv2.VideoCapture(1).read()**

**//Averaging Blur**

**blur = cv2.blur(img,(9,9))**

**// Conversion of BGR to HSV**

**hsv=cv2.cvtColor(blur, cv2.COLOR\_BGR2HSV)**

**// Providing Yellow coloured Frequencies**

**lb= np.array([15,80,50])**

**ub= np.array([39,255,255])**

**binary\_img = cv2.inRange(hsv, lb, ub)**

**// Final Output Image**

**cv2.imshow("output", binary\_img)**

**Execution Of Colour Detection:**

**Fig 1:Captured Image**



**Fig 2: Blurred Image**

****

**Fig 3: Binary Image**

We couldn’t use only colour detection mechanism for the task because it will not give the perfect output for cascaded stone detection instead it will detect other objects near which are in yellow coloured frequencies.