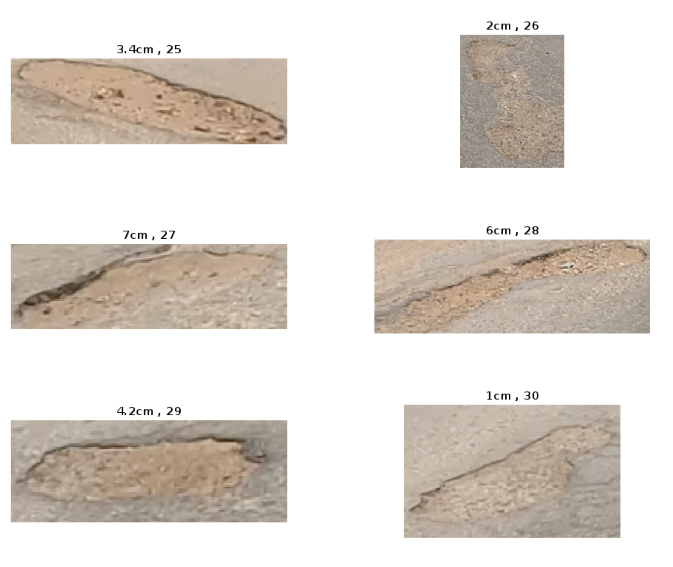
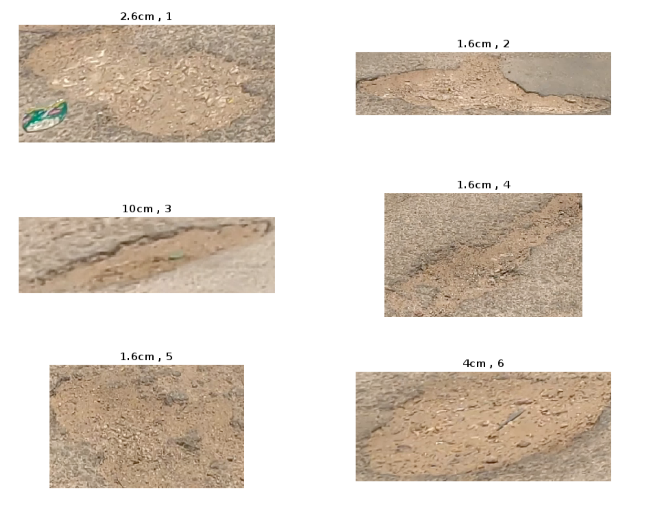
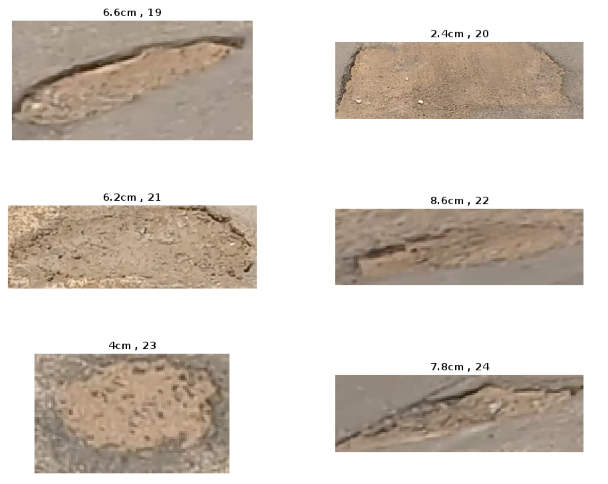
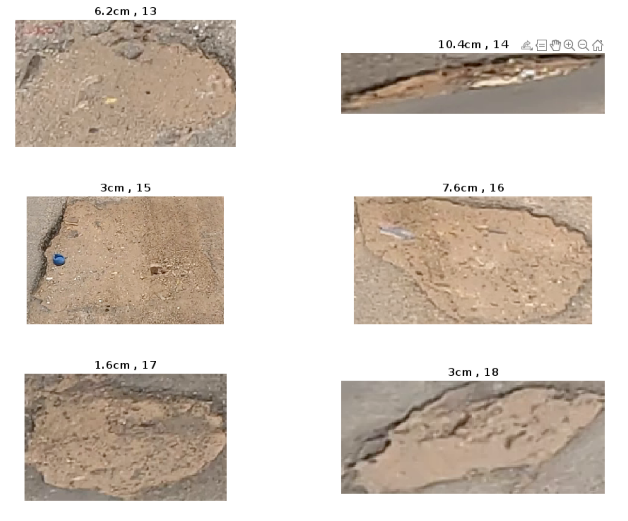
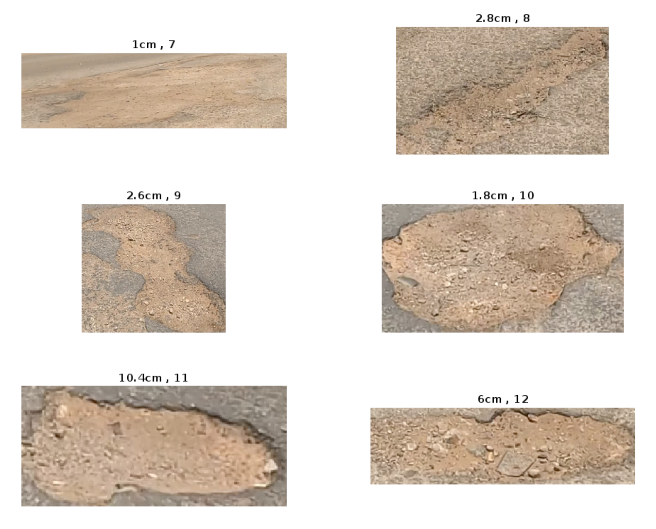
**SmartaThon**

**Pothole-Depth - :**

**This is the estimated depth in cm, and the id of the photo, using very low cost approach.**

****

****

**After obtaining the pothole from the bounding box applied by the model,**

**Feature extraction process has been done.**

* **First is to take the difference between the color of the edges and the pothole.**

****

Figure 1

* **Second is binarize the photo to obtain edge**

**A picture containing text, weapon

Description automatically generated**

Figure 2

**These two features are calculated and combined to be a score of how worse the pothole is.**

**This is the sorted array carrying the ids of the potholes from the shallowest to the deepest:**

**[7 30 2 4 5 17 10 26 20 1 9 8 15 18 25 6 23 29 12 28 13 21 19 27 16 24 22 3 11 14].**

**After sorting the values it got multiplied by a constant make it close to the cm representation (eg. score is 52, cm is 10.2)**

 A picture containing ground, outdoor, sandy, shore

Description automatically generated

Deepest shallowest

**After that the binarized image is used to get the area of the pothole, the area was calculated by referencing to an item found in the 34rd image in scene 2. The item is a cap for bottle of drinking water. The cap has a standard area of 28.27.**



bottle cap

**Technically:-**

**As shown in figure 1 the photo is divided into 5 boxes and gray scale of it is obtained,**

**Each box is summed and divided by the size of it.**

**The corner boxes is summed and divided by 4 and the difference between the boxes and the box in the middle is calculated.**

**The score would be ranged form 0 to 255 , 255 is to fully being different**

box1=img(1:10,1:floor(c/3));

box1=sum(sum(box1))/sb;

box2=img((r-9):r,1:floor(c/3));

box2=sum(sum(box2))/sb;

box3=img(1:10,(floor(c/3)\*2):c);

box3=sum(sum(box3))/sb;

box4=img((r-9):r,(floor(c/3)\*2):c);

box4=sum(sum(box4))/sb;

box5=img((floor(r/2)-5):(floor(r/2)+4),floor(c/3):(floor(c/3)\*2));

box5=sum(sum(box5))/sb;

**This score is added to the value of how much refine the edge is, this obtained by using this pipeline**

* **Binarize photo.**
* **Using median filter that its size is varying in respect to the pothole size.**
* **using bwlabel() method in matlab.**

**Code:**

Score2 = imbinarize(img);

Score2 = medfilt2(Score2,[di,di]);

[l, n] = bwlabel(Score2);

**n is how many object detected, if it is low that means the edge is refining.**

**Because n range from 0 to infinity this score must be terminated, 450 object detected is no difference than 70, all of it means that there is no refine edge.**

**So if n is larger than 30 it is no longer important so it set to be 0.**

**If n is ranging between 1 and 5, n is set to be 30. So that the bigger n the more deep the pothole is.**

if n == 0

n = 0

end

if n >30

n=0;

elseif n>20

n=2;

elseif n >10

n = 5;

elseif n>5

n=15;

elseif n>0

n=30;

end

**to find the area, bwarea() method is used to get the number of pixels, the method count the white pixel so the negation  operation is needed. The image of cap is loaded after cropping it to get size of cap only.**

a = bwarea(~Score2);

area = a \* 28.27 / (size(cap\_img,1)\*size(cap\_img,2));