import numpy as np  
import math  
import cv2  
 # we will read the image unchanged  
img = cv2.imread('field 1-2.jpg', cv2.IMREAD\_UNCHANGED)   
roi = img[600:1800, 50:440]  
roi = cv2.GaussianBlur(roi,(3,3),0)  
roi2 = roi.copy()  
roi = cv2.cvtColor(roi, cv2.COLOR\_BGR2GRAY)  
roi = cv2.Canny(roi,100,550,3)  
  
pixel\_to\_metric = 134.0596881  
initial\_pole = 464.0387914819  
intial\_angle = 89.26  
lines = cv2.HoughLinesP(roi,1,np.pi/180,100,maxLineGap = 100) #Hough transformation.   
no\_of\_lines = len(lines)  
  
edges = [] #LINE SEPERATION  
angles= []  
lengths = []  
for i in range(no\_of\_lines):  
 if ((lines[i][0][3]-lines[i][0][1]) != 0 and (lines[i][0][2]-lines[i][0][0]) != 0):  
 ang = abs(math.degrees(math.atan((lines[i][0][3]-lines[i][0][1])/(lines[i][0][2]-lines[i][0][0]))))  
 if (abs(90 - ang) <= 5):  
 edges.append(lines[i][0])  
 angles.append(ang)  
 lengths.append(math.sqrt((lines[i][0][2]-lines[i][0][0])\*\*2 + (lines[i][0][3]-lines[i][0][1])\*\*2))  
 elif ((lines[i][0][2]-lines[i][0][0]) == 0):  
 edges.append(lines[i][0])  
 angles.append(90)  
 lengths.append(math.sqrt((lines[i][0][2]-lines[i][0][0])\*\*2 + (lines[i][0][3]-lines[i][0][1])\*\*2))  
temp=angles.copy()  
  
for i in range(len(edges)): #LINES DRAWN  
 cv2.line(roi2, (edges[i][0],edges[i][1]), (edges[i][2],edges[i][3]), (0,255,0), 1)  
  
index = [angles.index(max(angles))] # 2 LINES CLOSER TWO 90 DEGREES   
temp[index[0]] = 0  
index.append(temp.index(max(temp)))  
  
 # START AND END POINTS  
if (abs(lengths[0]-lengths[1]) <= 20):   
 start\_point = (int((edges[index[0]][0]+edges[index[1]][0])/2), int((edges[index[0]][1]+edges[index[1]][1])/2))  
 end\_point = (int((edges[index[0]][2]+edges[index[1]][2])/2), int((edges[index[0]][3]+edges[index[1]][3])/2))  
else:  
 start\_point = (edges[lengths.index(max(lengths))][0],edges[lengths.index(max(lengths))][1])  
 end\_point = (edges[lengths.index(max(lengths))][2], edges[lengths.index(max(lengths))][3])  
cv2.line(roi2, start\_point, end\_point, (0,0,255), 1)  
  
 # SNOW LEVEL CALCULATIONS.  
pole\_length = ((start\_point[0]-end\_point[0])\*\*2 + (start\_point[1]-end\_point[1])\*\*2)\*\*0.5  
snow\_level = initial\_pole - pole\_length  
snow\_level\_metric = snow\_level/pixel\_to\_metric #angle of the pole is neglected because only edges with 90 +- 5 is taken and in worst case sin(85) = 0.99  
snow\_level\_metric = round(snow\_level\_metric,5)  
print("snow\_level\_metric:",snow\_level\_metric)  
  
cv2.putText(roi2, str(snow\_level\_metric), (30,434), cv2.FONT\_HERSHEY\_SIMPLEX, 0.6, (0,255,0), 2)  
cv2.putText(roi2, "mts", (30,454), cv2.FONT\_HERSHEY\_SIMPLEX, 0.6, (0,255,0), 2)  
  
cv2.namedWindow('roi',cv2.WINDOW\_NORMAL)  
cv2.imshow('roi',roi2)  
  
cv2.waitKey(0)  
  
cv2.imwrite("field 1-2 edited.jpg", roi2)  
cv2.imwrite("field 1-2 binary.jpg", roi)