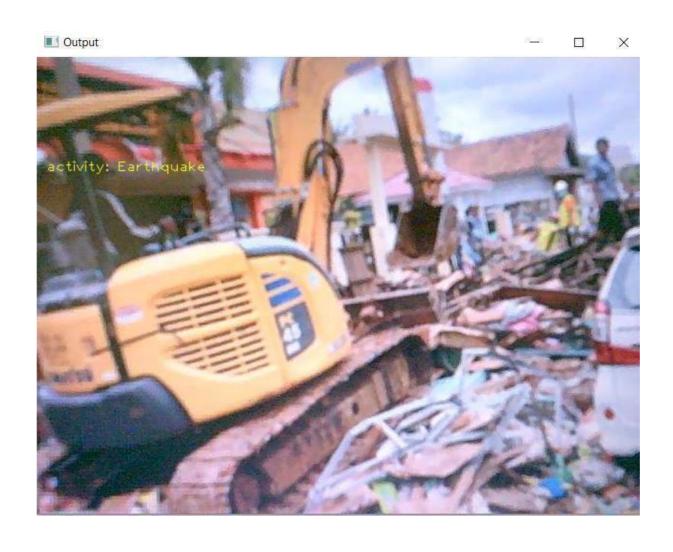
## PROJECT DEVELOPMENT PHASE

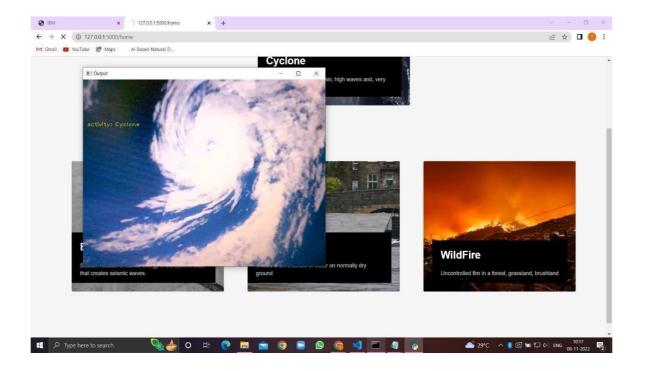
## **SPRINT-4**

## INTEGRATE THE WEB APP WITH AI MODEL:

After creating the Model, the Model should be integrated with the web app using the Flask application. The coding part is named as app.py and it will be running in the localhost through the generated link. By navigating the localhost the webpage will be visible.

```
output = frame.copy()
74
                 #print("apple")
75
                 frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
76
                 frame = cv2.resize(frame, (64, 64))
                 #frame = frame.astype("float32")
77
                 x=np.expand_dims(frame, axis=0)
78
                 result = np.argmax(model.predict(x), axis=-1)
index=['Cyclone','Earthquake','Flood','Wildfire']
79
30
                 result=str(index[result[0]])
31
32
                 #print(result)
                 #result=result.tolist()
33
34
35
                 cv2.putText(output, "activity: {}".format(result), (10, 120), cv2.FONT HERSHEY PLAIN,
36
                                1, (0,255,255), 1)
                 #playaudio("Emergency it is a disaster")
cv2.imshow("Output", output)
key = cv2.waitKey(1) & 0xFF
37
38
39
90
91
                     ## if the `q` key was pressed, break from the loop
                 if key == ord("q"):
                      break
93
94
95
            # release the file pointers
            print("[INFO] cleaning up...")
96
             vs.release()
97
98
             cv2.destroyAllWindows()
            return render template("upload.html")
99
90
31
  if __name__ == '__main__':
32
          app.run(debug=False,threaded=True)
33
34
```





## **REGISTER FOR IBM CLOUD:**

Registering IBM Cloud for deploying the model through the Flask application. The IBM Cloud account is created for all the team members.

