

geoScope

By Code Crushers



Architecture

Used Flask for building the webApp.

Used openCV for facial recognition.

Used Azure for hosting.

Used Docker for deployment.

How does it work?

One of the most important aspects of managing a natural disaster is to identify the people affected by it. Our app will create a centralised database that will enable the relatives as well as relief teams dispatched by the government to conduct a crowd sourced, real time census of people affected by a natural disaster.

The rescue teams will upload facial photos of the people(single face per image) rescued along with any personal information they can obtain like name, mobile number, date of birth, age, and address. They will then append information like the coordinates of where the said person was rescued from, the status of his injury, and the relief camp/hospital he was taken to.

How does it work?

Now, there is a high possibility that the person is either unconscious, dead, or unable to respond to questions. This is where the facial recognition part will step in. The friends, neighbours, and relatives of that person can upload a single picture on our server.

We will then match that information with the available pictures in our database (using One Shot Learning). If a match is found, the information about that person will be returned. People who have been taken to the relief camps can further provide information about their neighbours and the people that they know. All the information will be aggregated and duplications merged.

Ps: We used a pre-trained model for this. (ResNET)

How does it work?

This way, an accurate estimate of the affected population can be identified using crowd sourced information which will greatly help the relief teams in planning out their rescue operations.

People who are stranded can also mark their locations on a map along with their essential information like picture, name, age etc. This information will be made available to the relief teams(volunteers, army, navy), who can then take better decisions based on the population density in different regions made available on a map using the Bing Maps API.

Thanks!

