CITIZEN SERVICE PROBLEM (CSP)

[1]Aditya Kapadia, [2]Lucky Rathod and [3]Jigar Panchal
Dwarkadas .J Sanghvi College of Engineering,
Vile Parle (W), Mumbai 400056
Email:[1]aditya.kapadia5@gmail.com,
[2]luckyrathod46@gmail.com,[3]panchaljigar82@gmail.com.

Domain: Artificial Intelligence.(Neural Network)

Abstract:

In the recent monsoons, due to continued waterlogging, potholes wrenched havoc in the daily lives of people. From traffic jams to injuries caused simply due to walking through waterlogged roads, all this can primarily be attributed to improper garbage disposal and the pothole pandemic. If there were an easy means to tackle this issue, majority of the problem faced during heavy rains could be averted. Our application is specially designed to handle this large-scale garbage and pothole pandemic using our own custom built convolutional neural Network. An application that can do everything, from capture, to report and classify, to follow through, to tag on the map and even provide feedback.

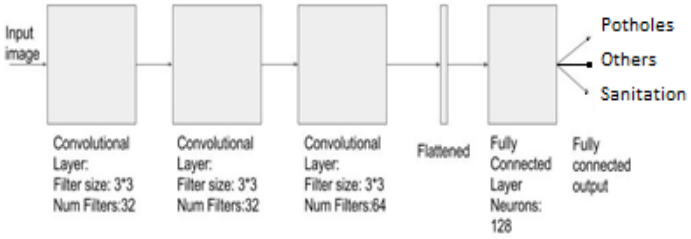
Keywords: Citizen, Problem, Potholes, Sanitation, Garbage, Convolutional Neural Network, OpenCV, numpy, tensorflow, firebase, Contour Detection

I. INTRODUCTION

Empowering Citizens is the need of the hour.

Necessary public issues related to sanitation, drinking water, clean and pothole free roads and clean public toilets to name a few, are not being addressed. Inadequate use of funds is the primary cause for this. Empowering citizens to identify issues notify the authorities and follow up with their progress would be useful in tackling such issues, as India is a population/labor driven country. The authorities can then be held accountable for the same. Regular follow-ups by citizens will ensure adequate results.

Our Solution enables citizens to report issues and follow up those problems until resolution is done. Citizens will report the problems by clicking the photograph of a pothole or sanitation related issue. With this the location and the approximate specifications of the issue, namely depth and width for potholes, and amount of garbage for sanitation will be determined. A convolutional neural network created by us does identification of the specifications of the pothole and sanitation issue.



II. MODULAR DESCRIPTION AND ARCHITETCURE

A. Android App

1. **Pothole and Sanitation Issues:** When the citizen identifies an issue, he reports the issue using our app to a local NGO by capturing a photograph of the issue along with the location using geotagging. Our custom-built neural network will then classify the reported issue and gives specifications of the said issue. If a pothole, it will specify the width and height of the pothole and if it is a sanitation related issue, then it will give the total no. of objects present.
2. **Map:** The user can see all the globally reported issues, by anyone, anywhere and anytime on the map, which is, built-in the app itself. The map is completely real-time and will also give directions to go that issue.

B. Firebase Real-time database

1. **Cloud Storage:** This may be the greatest part of our project. Our whole project backend from user registration to issue solving, is done on the cloud and real-time. This makes our app

super-efficient and extremely light (only 4 Mb.)

C. Neural Network

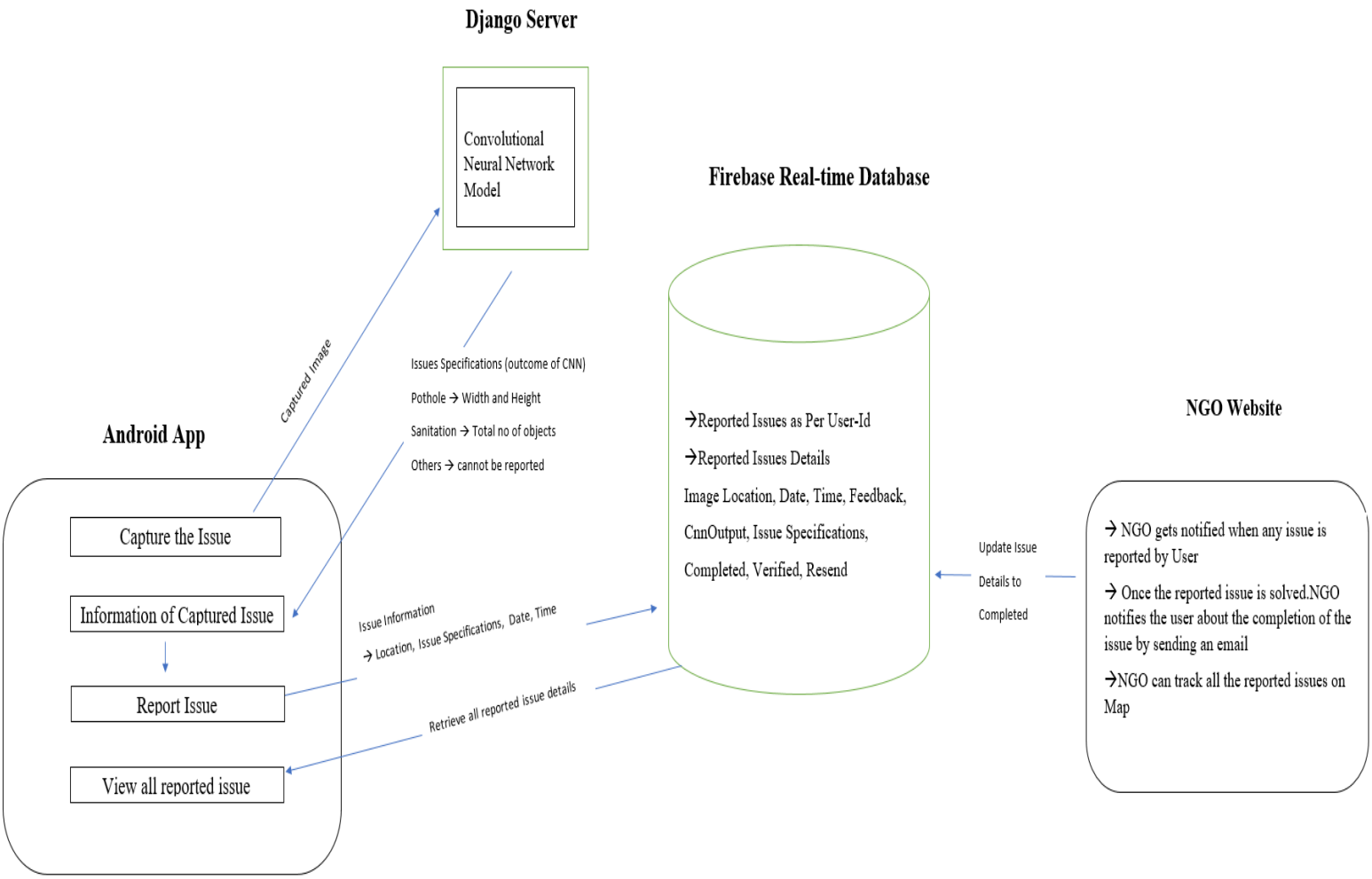
1. **Data Sets:** For the data set, we have captured as well as downloaded images of roads, construction site, etc for the training of the neural net. After capturing the images from various locations, we have extracted the main contents of pictures i.e., “potholes”, “sanitation” ,”Others” by cropping the images into 1:1 from all the three sides i.e. top-corner, centre , bottom-corner so that each image will be split into 3 sub-images. Next, we clean the unwanted data from the images and keep only the copy of the image in which the main data is contained. Finally we scale all the images into 160px*160px for better accuracy.

   These images (approx.10000 of them) are given to the neural net for training purposes to predict the data more accurately, precisely. In addition, as the user, keeps uploading images these are added to the training data and the above process from extracting, splitting to scaling is done once again, leading to a flawless neural network in the near future.

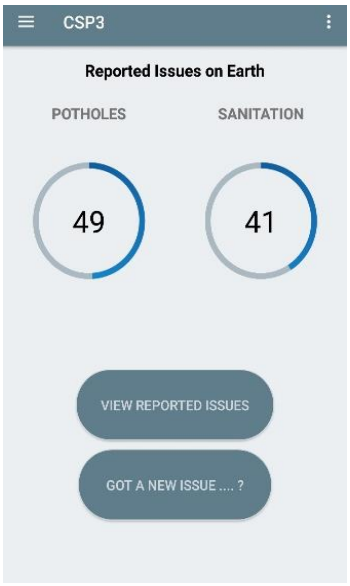
2. **Convolutional Neural Network (CNN) Architecture:** Using Tensorflow framework we created a state-of-the-art network, which uses three convolutional layers, one flattening layer and two dense layers. Input to the layers is an image of specification 160*160px. The input image is converted to a 2D vector. This vector is then reshaped and converted to a tensor to include its colour information. We are using red, green and blue coloured images.
   The first two convolution layers have 32 filters and the third one has 64 filters, used for extrapolating information from the image. Size of the filter is 3px*3px. Padding is used to keep dimensions of the output tensor equal to the input tensor.
   After passing through the convolution layers, the modified tensor then passes through a flattening layer to easy with prediction. The flattening layer converts the image to a 1D-feature vector. This vector is then passed to a dense/fully connected layer with 128 nodes. Finally we pass the output of the dense layer to the final dense layer which outputs the probability of the input image in terms of three fields, ‘Sanitation’, ‘Pothole’ and ‘Other’.
   Activation of the output layer is governed by a softmax function.
   To predict width of the input image, in the event it is a pothole, we have used OpenCV. To get accurate predictions our research stated that the input image needed to be 160*160px which also happened to be the input size to our network. Using OpenCV’s contour detection libraries, we were able to successfully predict the width of potholes.
   For sanitation we used the same 160*160px image input. Again using OpenCv’s contour detection libraries, we were able to successfully predict the number of objects in the image. Again our research dictated that the total contour length divided by 27 gave us a near accurate estimate of the number of objects in the image.

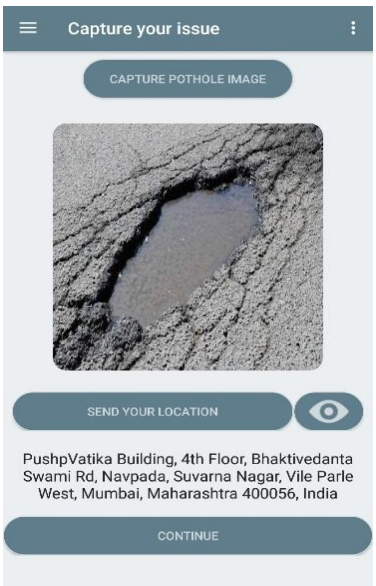
D. NGO Website

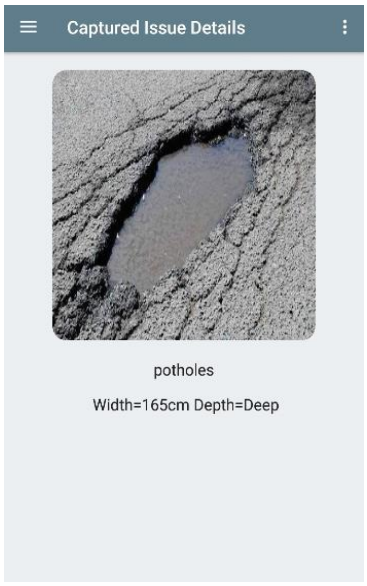
1. **Reported issue information:** The NGO can see all the issues reported by any user with its specifications. Now when the NGO solves any issue, he can press a simple button on the website, which will notify the user of that issue, regarding the completion of the problem.
2. **Feedback:** User can provide feedback directly to the NGO once the issue is solved, from the app, and the NGO can reply to the feedback given directly from the site. This helps in easy feedback and learn system which is quite beneficial for a good
3. **Map:** The user can see all the globally reported issues, by anyone, anywhere and anytime on the map, which is, built-in the app itself.

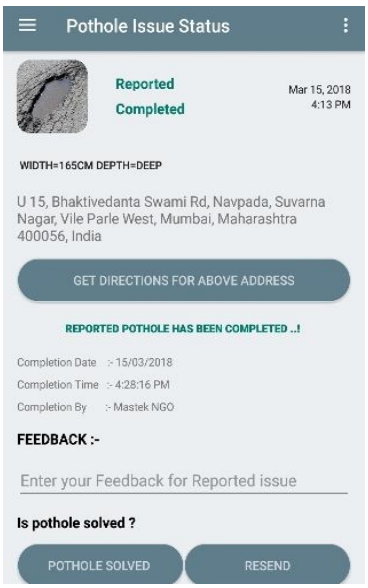


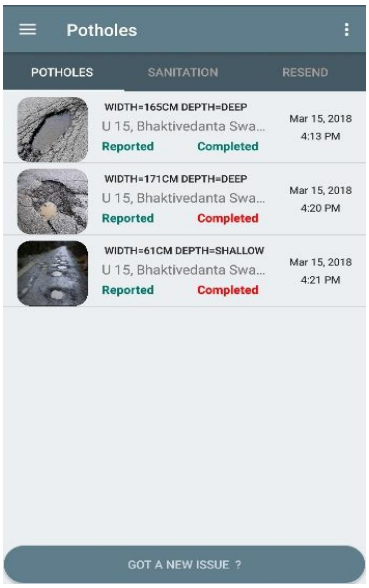
III. SCREENSHOTS OF CSP

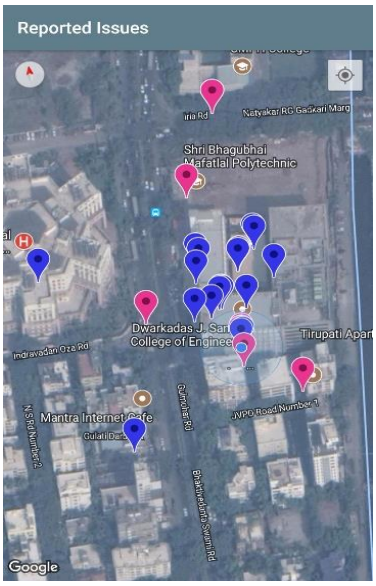












IV. CONCLUSION

An App in which a single touch can log your location data and display it too. State of the art custom built neural net to classify all your problems without you having to do so. A smart network that only gets smarter with time. A database so versatile, it even did away with SQL. All issues, images and follow through are updated everywhere as and when they occur. In short direct feedback, In Real Time. All this in an extremely small package on the cloud.

Our solution also provides the NGOs with a portal to monitor issues logged by the citizens everywhere. The people in charge can then address issues and update the citizens about progress regarding the same. This ensures involvement of all, the reporter and the solver.

V. REFERENCES

1)Bulk image downloader by Martin Brinkmann
2)www.ghacks.net
3) https://imageresize.org/bulk-resize
4) www.cv-tricks.com
5)https://docs.aws.amazon.com/machine-learning/latest/dg/training-process.html
6) https://www.tensorflow.org
7)https://www.tensorflow.org/get_startd
8) https://www.tensorflow.org/serving/
9) http://php.net/docs.php
10)https://firebase.google.com/docs/guides/
11)https://firebase.google.com/docs/web/setup
12)https://developer.android.com/guide/index.html