Project Title: Jyoti – Application that helps blind and dumb receive basic elementary education.
Description: 5th Semester project on Machine Learning using Python for CSE581, Software Application Laboratory.
Institute: University Institute of Technology, The University of Burdwan.

Project Abstract:

The project demonstrates a working application that helps victims of visual impairment, dumbness or victims suffering from both receive elementary education. “Subjects” in this document will hereby mean victims of visual impairment, dumbness or both. The project aims to provide all round development for subjects using technological advancements made in the field of cloud and machine learning. Our innovative project helps subjects undergo self assessment tests and general knowledge lessons seamlessly without any wearable equipment. This makes education for subjects interactive and as engaging as a normal student gains at his/her school.

Prerequisites:

The project assumes knowledge of the following resources:

1. Programming Languages: Python, Java, Node.js, HTML, CSS, JavaScript.
2. Database: Firebase Realtime Database.
3. Frameworks used: Maven (Java), Swing (Java), Android (Java), Gradle, Express, Materialize, Twitter BootStrap.
4. Cloud Services used: Firebase Hosting, Google Cloud Platform (Google Cloud Functions), IBM Cloud Cognitive Services.
5. ML Libraries used: Tensorflow, OpenCV, Numpy, Pyaudio.
6. Courses undertaken: Video analytics using OpenCV and Python shells [1], Introduction to Machine Learning.

Related Works:

The project work involved several experience with projects involving cloud hosting, appropriate references are provided below:

1. Cloud Technologies: TeamSpeak Application [2]
2. Related works on Java technology: ServerAnalyser [3], Sirena [4]
3. Datasets: Hand Gesture Dataset [5], Inception Dataset [6][7]

Overview of Machine Learning and Tools used:

The Jyoti project demonstrates three applications. They are:
1. Native Client Application
2. Android Application
3. Web services for native client application services.

All three applications have importance in their respective domains and a detailed discussion regarding each is presented below.

1. Native Client Application

   The Native Client application can be divided into four modules, namely Graphical User Interface (GUI), Database Service, Authentication Service and Machine Learning services. The overall project is packaged under a JAR file with necessary dependencies.

a. *Graphical User Interface*: The GUI of the application is made possible using Java Swing framework for most of the application and a part of it achieved using cv2 windows in Python (during gesture recognition).
b. *Database Service*: The database service uses cloud technology to store user's score data and quiz dump on the cloud so that it can be accessed from any device with ease. Also, the data will be available 24 x 7.
c. *Authentication Service*: The Jyoti Native application uses Firebase Authentication Service from Google which prevents unauthorised access to a user's account unlike MySQL databases prone to SQL injections and other vulnerabilities.
d. *Machine Learning Service*: The native application uses machine learning service by invoking a native python script under *scripts/* directory. This script has runs the gesture recognition module and returns detected fingers/frame data to *BufferedInputStream* inside Java application. The application receives all inputs, computes an average of the value, restricting it between 1 and 3 (since, three options are provided upon which selection is to be made). The Java application also embeds an IBM Cognitive Services SDK (*com.ibm.watson.developer_cloud.text_to_speech*) for converting text to voice responses. It uses an internal authentication service known as IAM (Identity and Access Management) Authentication (*com.ibm.watson.developer_cloud.service.security.IamOptions*) to access IBM Cloud's services.

2. Android Application

The Android application of the Jyoti project is written in Android Java using the following *gradle* modules:

```
implementation 'org.tensorflow:tensorflow-android:+'
implementation 'com.android.support:appcompat-v7:27.1.1'
implementation 'com.android.support:support-v4:27.1.1'
implementation 'com.android.support:design:27.1.1'
implementation 'com.squareup.okhttp3:okhttp:3.12.0'
implementation 'com.google.code.gson:gson:2.8.5'
```

Additional improvements to the training model can be made by configuring the *nativeBuildSystem* to *'bazel', 'cmake', 'makefile'*. However, we tested out the results with *nativeBuildSystem* none on a Windows System and the results are appreciable with objects being detected with an accuracy upto 99% for a camera resolution of 640 x 480. The trained model is available under *assets/* directory named as tensorflow_inception_graph.pb.

3. Web services for native client application services

The web service that powers the quiz engine of the Jyoti native application is built using Node.js, Express server that hosts a website from where the client JAR application can be downloaded. However, the web server is not a static server, rather a dynamic API service that has the following endpoints:

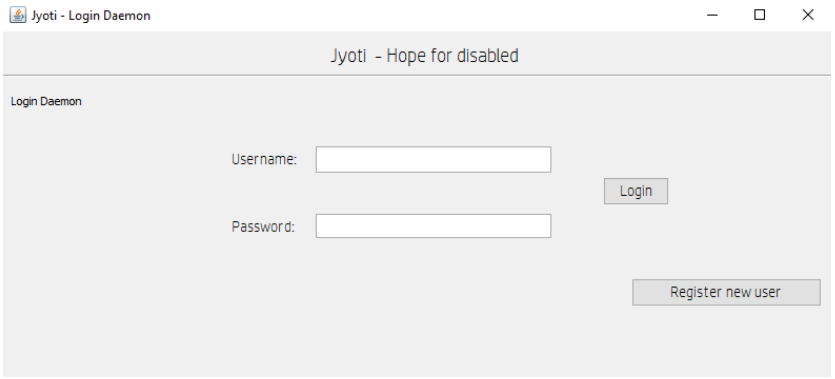
a. https://jyotiserver.firebaseapp.com
This endpoint hosts the static website from where the Java JAR application can be downloaded. The frontend of the website is designed using HTML, CSS, JavaScript, Materialize and Twitter BootStrap.

b. https://jyotiserver.firebaseapp.com/quiz.json
This endpoint is responsible for delivering an API service to the native client application. The data from the Firebase Realtime Database is retrieved and served dynamically using this service. It also has the ability to cache its content across different locations depending on their use. This will enable faster API calls and lesser load on the API Server thus, meeting the demands of scalability. The Server is hosted using Node.js 6 on Firebase Hosting and the API server running on Google Cloud Platform (Google Cloud Functions).

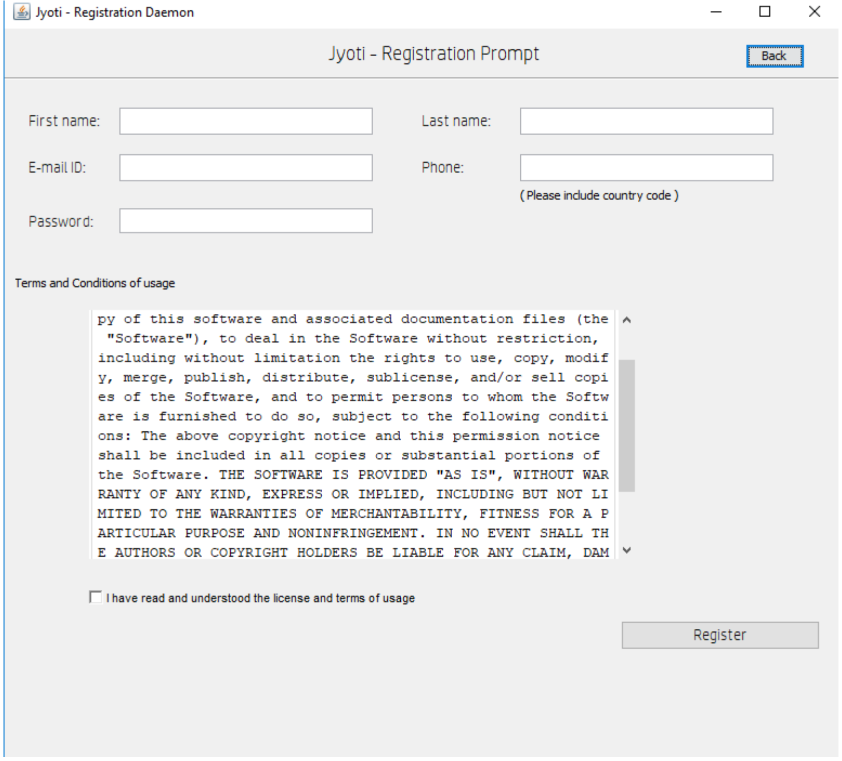
Screenshots of relevant services:

1. Jyoti Native Application

Login Window:

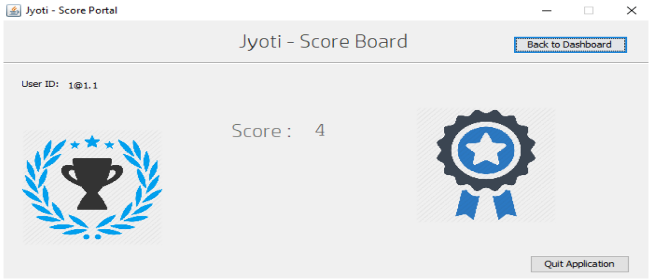


Register Window:


Jyoti - Registration Daemon
Jyoti - Registration Prompt
Back
First name:
Last name:
E-mail ID:
Phone:
(Please include country code)
Password:
Terms and Conditions of usage

```
py of this software and associated documentation files (the
 "Software"), to deal in the Software without restriction,
including without limitation the rights to use, copy, modif
y, merge, publish, distribute, sublicense, and/or sell copi
es of the Software, and to permit persons to whom the Softw
are is furnished to do so, subject to the following conditi
ons: The above copyright notice and this permission notice
shall be included in all copies or substantial portions of
the Software. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WAR
RANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LI
MITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A P
ARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL TH
E AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAM
```

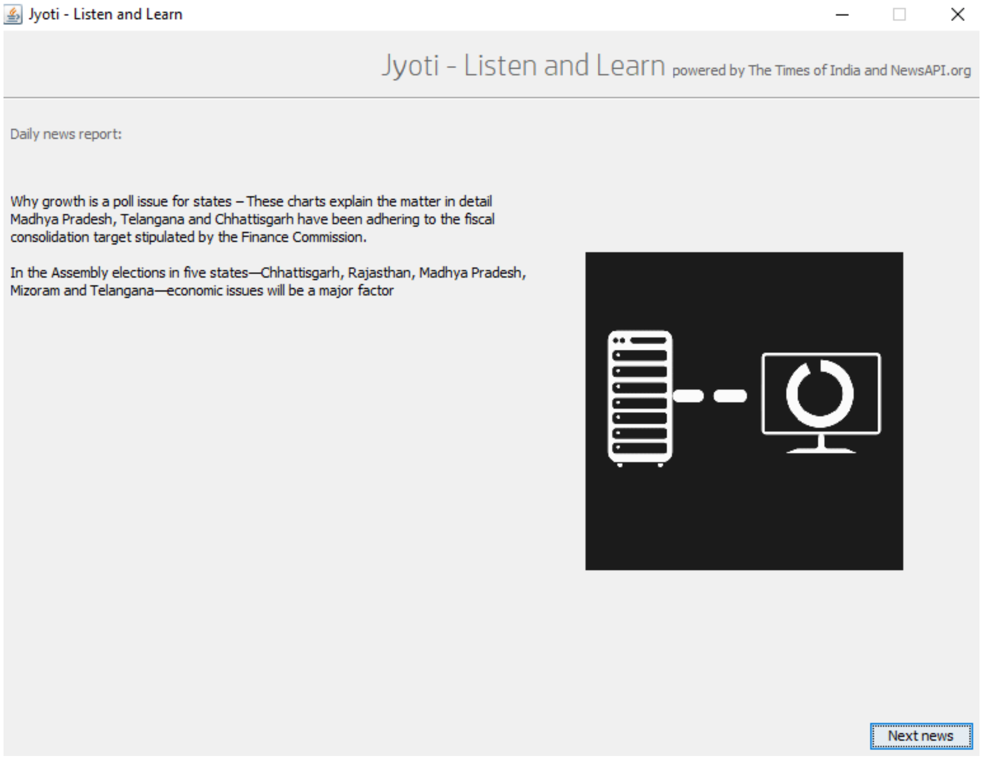
Score Window:



User ID: 1@1.1

Score : 4

Listen and Learn Service Window:



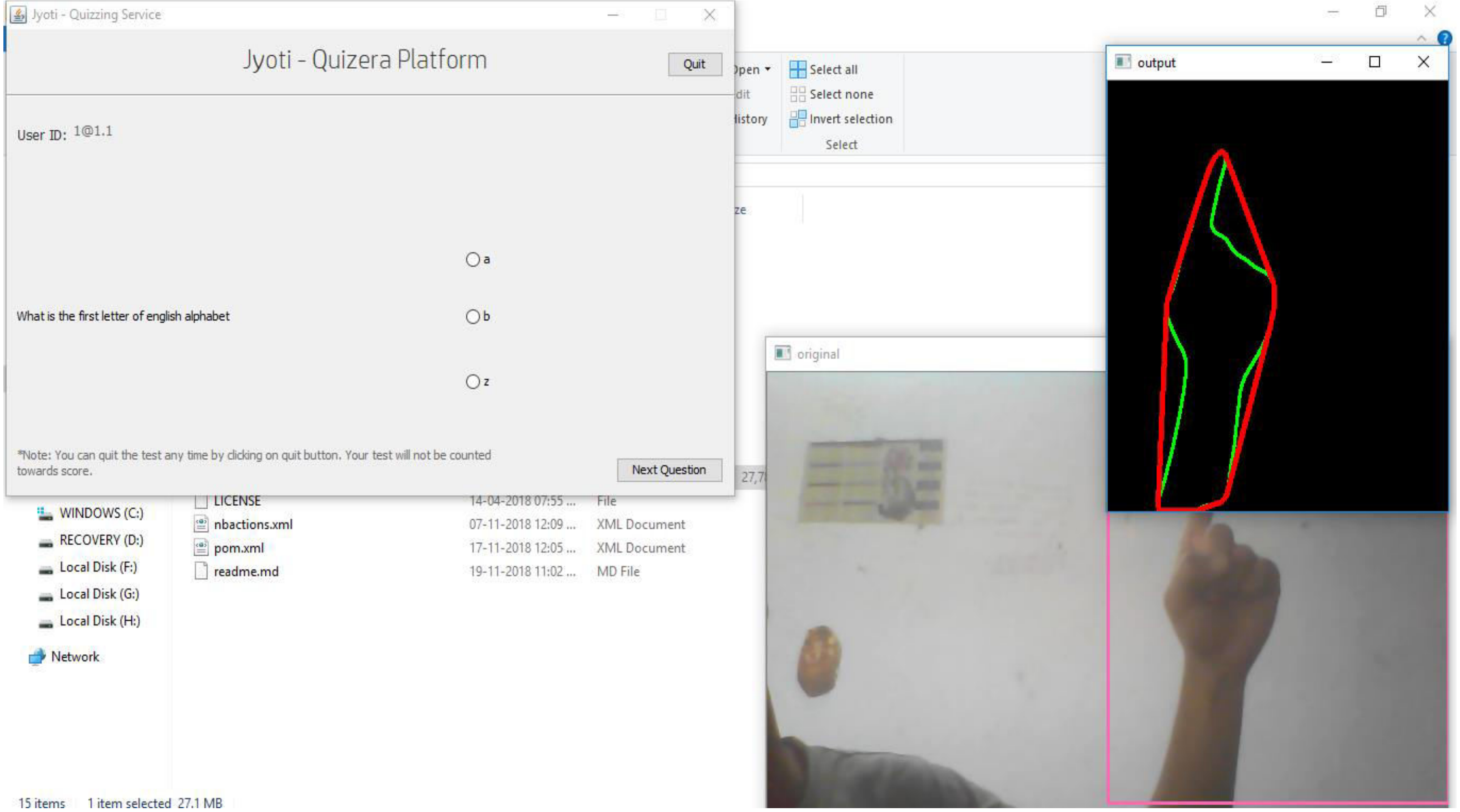
Jyoti - Listen and Learn powered by The Times of India and NewsAPI.org

Daily news report:

Why growth is a poll issue for states – These charts explain the matter in detail
Madhya Pradesh, Telangana and Chhattisgarh have been adhering to the fiscal
consolidation target stipulated by the Finance Commission.

In the Assembly elections in five states—Chhattisgarh, Rajasthan, Madhya Pradesh,
Mizoram and Telangana—economic issues will be a major factor

Jyoti Quizera Service Window:



Jyoti - Quizera Platform

User ID: 1@1.1

What is the first letter of english alphabet

○ a

○ b

○ z

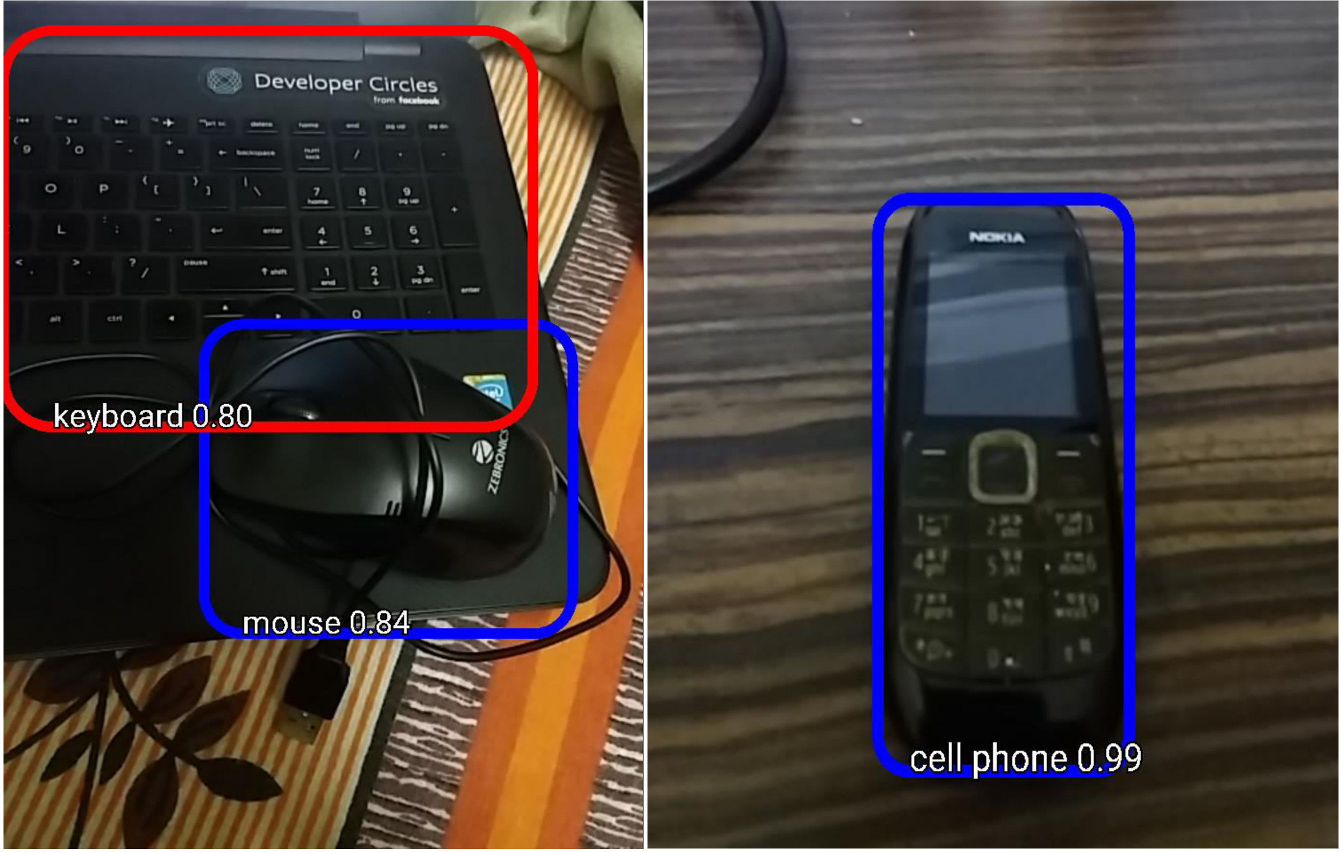
*Note: You can quit the test any time by clicking on quit button. Your test will not be counted towards score.

LICENSE	14-04-2018 07:55 ...	File
nbactions.xml	07-11-2018 12:09 ...	XML Document
pom.xml	17-11-2018 12:05 ...	XML Document
readme.md	19-11-2018 11:02 ...	MD File

15 items 1 item selected 27.1 MB

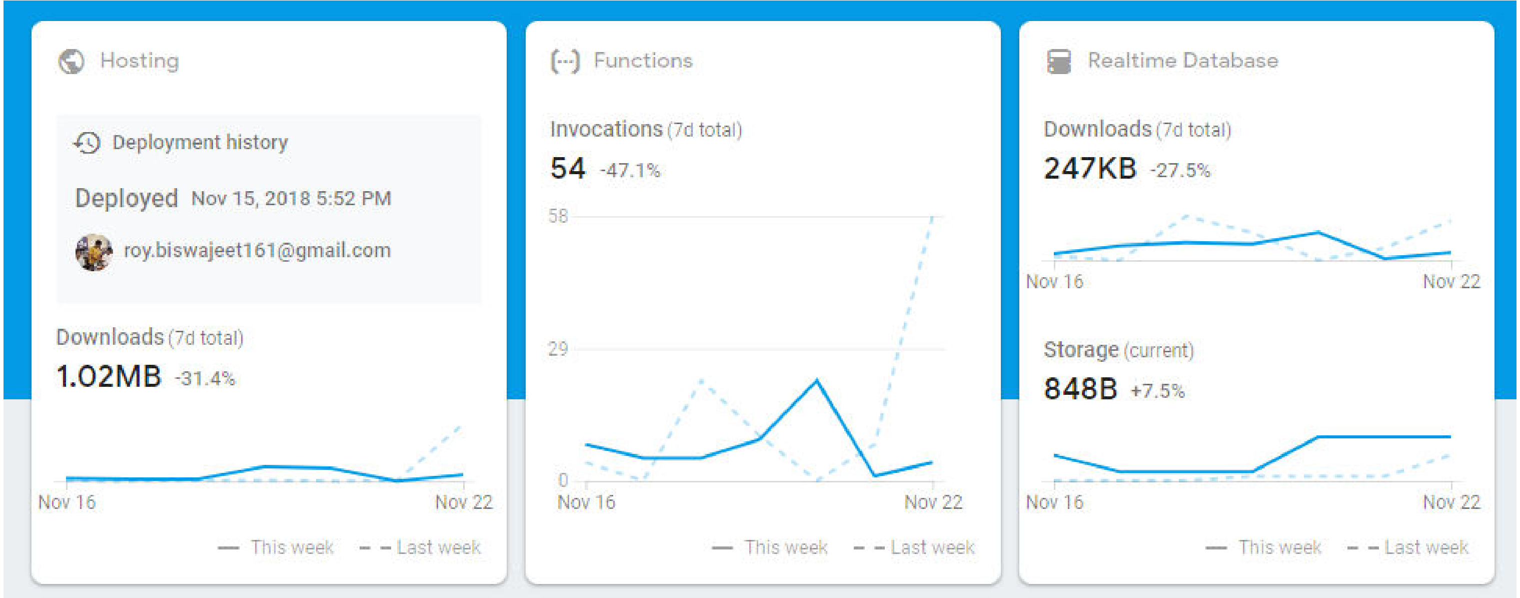
2. Android Application

Detecting objects from frames using the android application.



3. <u>Web Services</u>

Usage metrics of various cloud services:



Hosting

Deployment history

Deployed Nov 15, 2018 5:52 PM

roy.biswajeet161@gmail.com

Downloads (7d total)

1.02MB -31.4%

Functions

Invocations (7d total)

54 -47.1%

Realtime Database

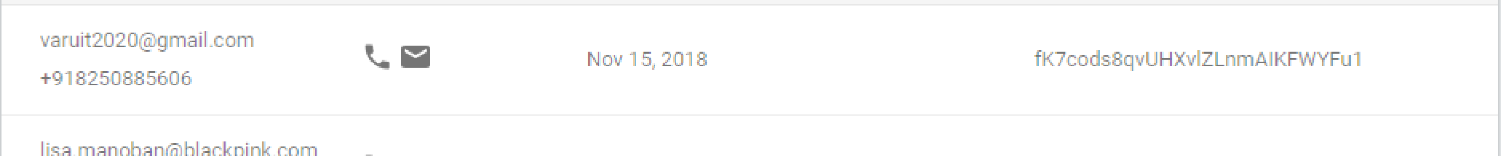
Downloads (7d total)

247KB -27.5%

Storage (current)

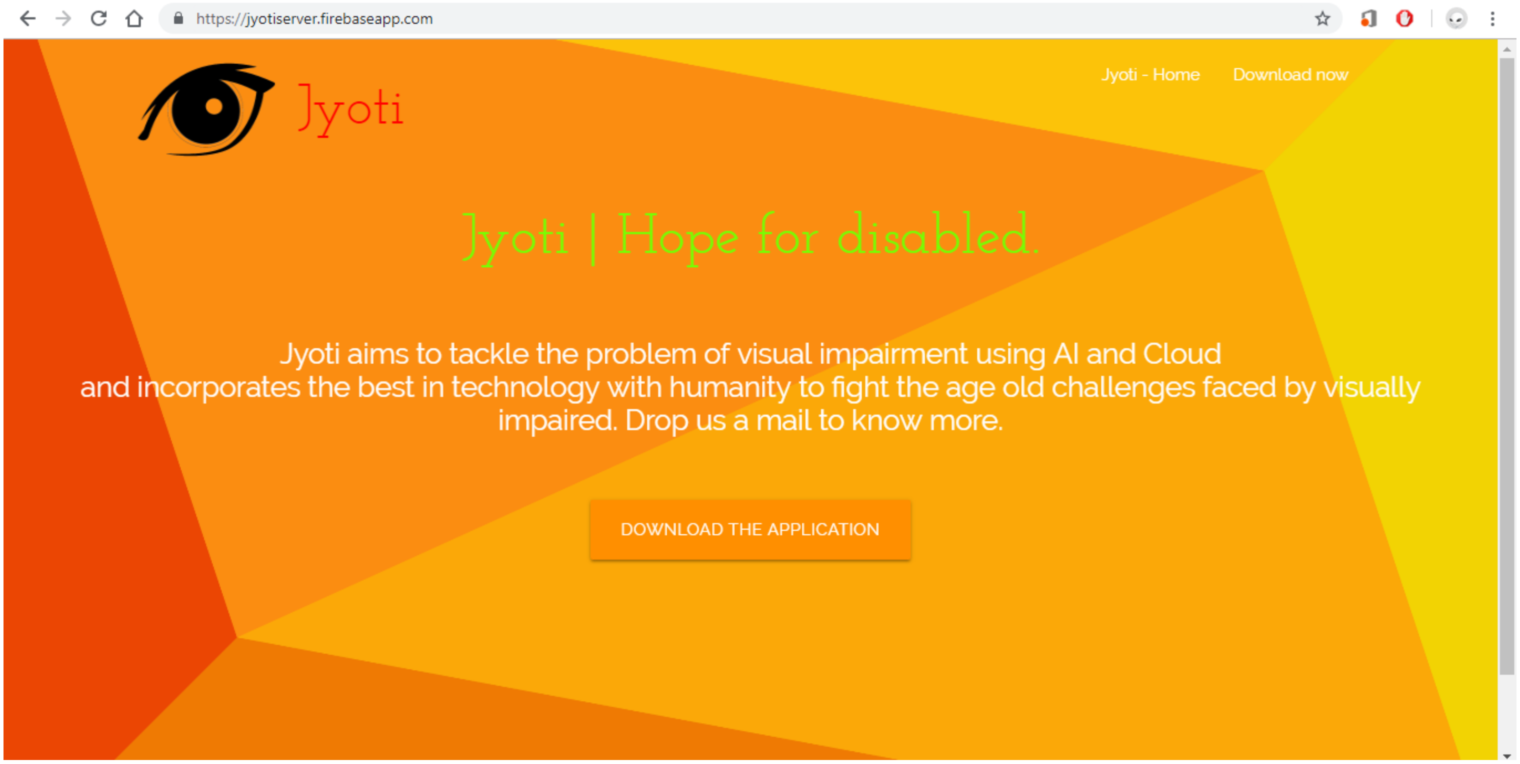
848B +7.5%

Data as seen on Firebase Authentication Console:



varuit2020@gmail.com +918250885606	☎ ✉	Nov 15, 2018	fK7cods8qvUHXvlZLnmAIKFWYFu1
lisa.manoban@blackpink.com	-		

Website:

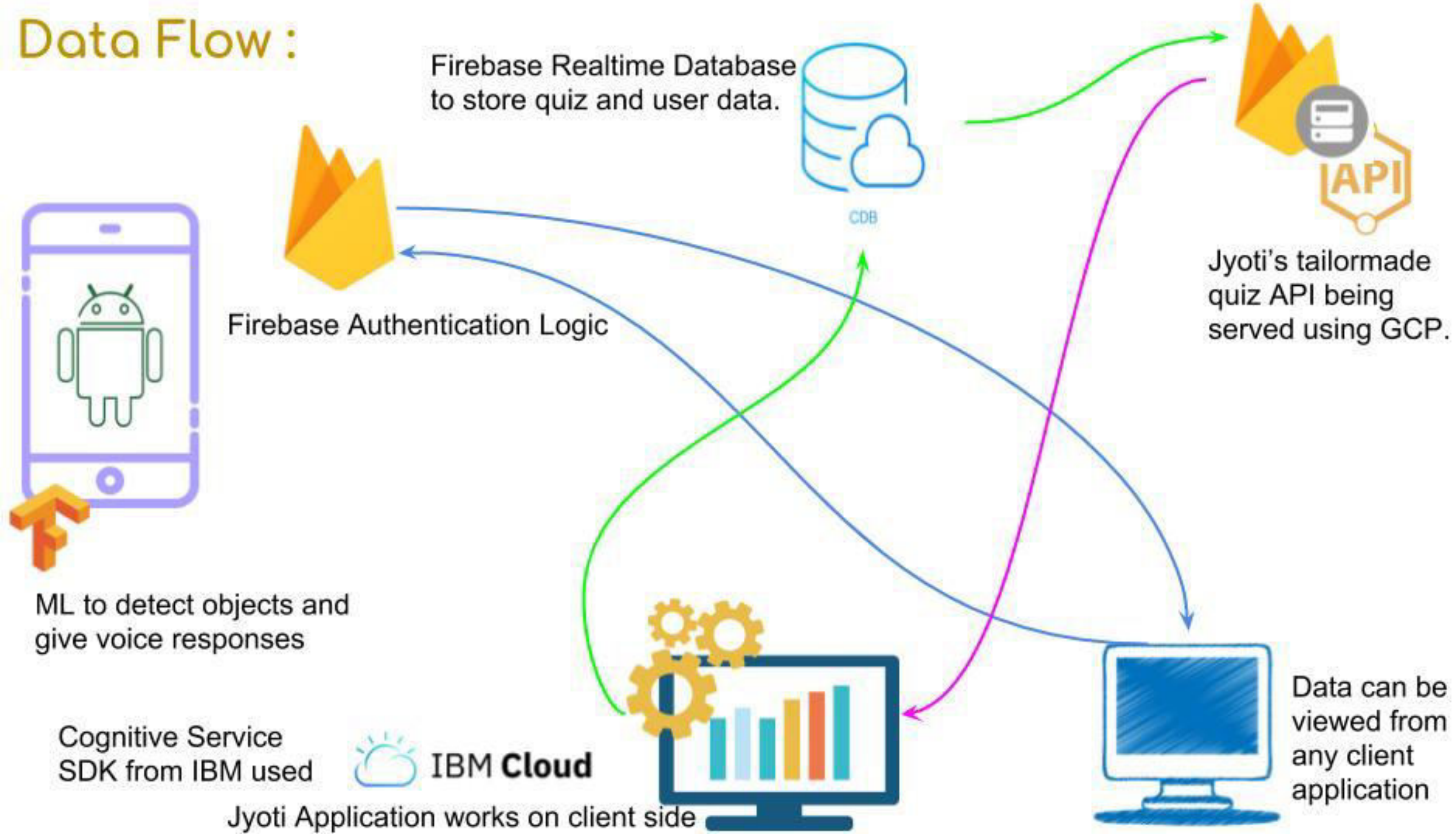


https://jyotiserver.firebaseapp.com

Jyoti | Hope for disabled.

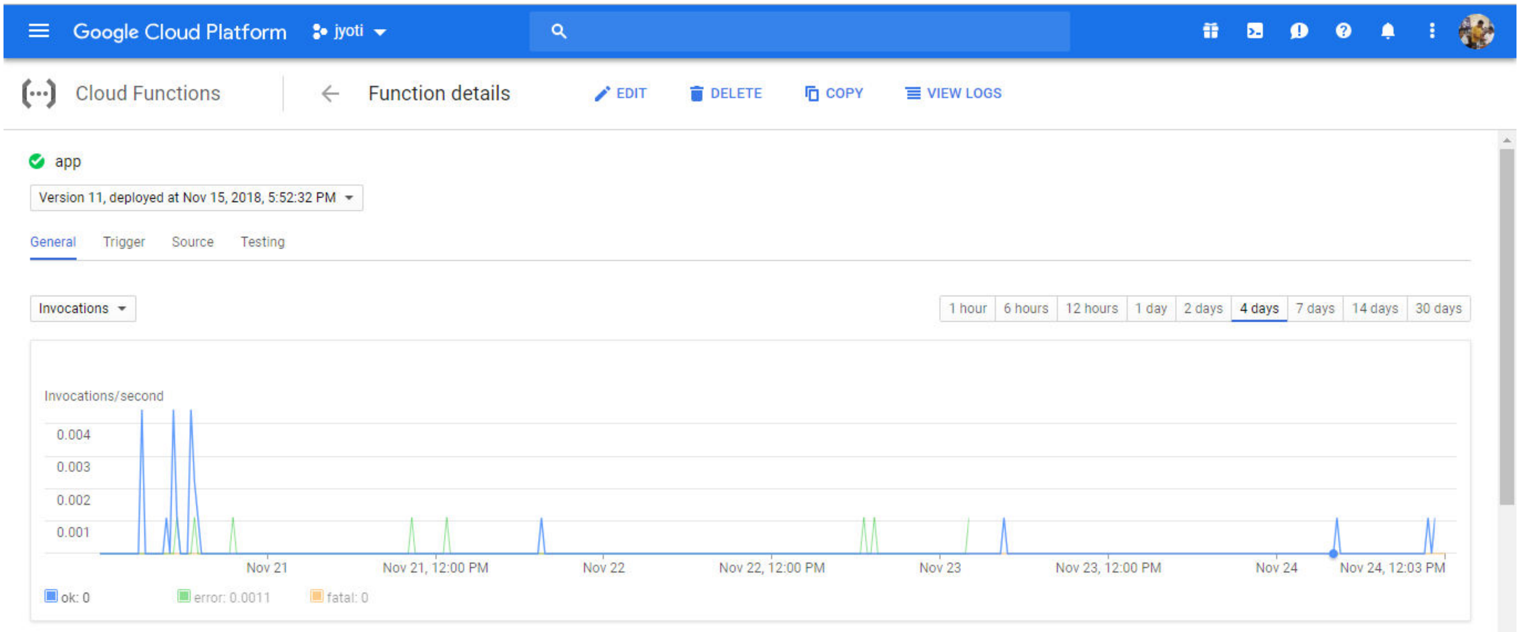
Jyoti aims to tackle the problem of visual impairment using AI and Cloud and incorporates the best in technology with humanity to fight the age old challenges faced by visually impaired. Drop us a mail to know more.

Algorithm/Flowchart:



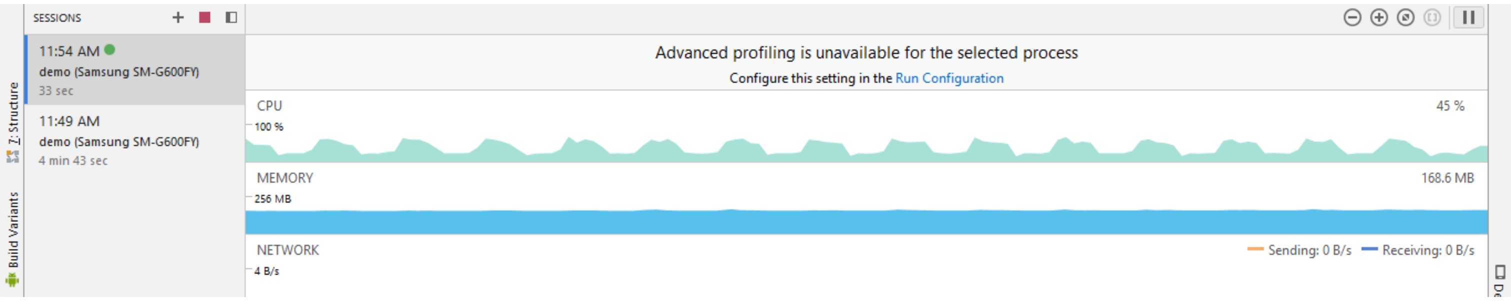
Results obtained showing Accuracy and Performance

The tailor made web API being served using Firebase Cloud Functions also has a unique feature. All API requests are made just only one per session to download the JSON data from the server for processing in the client application. This is done taking in care to develop the solution for third world countries like India where internet availability is provided at an appreciable cost and to lessen unnecessary API requests, we have optimised our application to make just one API request per session. Also, considering the network latency in remote cities and villages, we have enabled cache control services on our API, so whenever a user accesses API service, the response gets cached for 10 minutes in the nearest server to allow for another user to fetch the same resource faster.



Version 11, deployed at Nov 15, 2018, 5:52:32 PM

The android application’s CPU usage per session ranges around 35 – 45% and memory usage with an average of 168 MB with no network traffic since we are not setting up any authentication for our service and not monitoring the application history as it will only affect the application’s performance and cause hindrance in operation.



Advanced profiling is unavailable for the selected process

Configure this setting in the Run Configuration

Application of the proposed work

The proposed work can be made available on following areas:

1. Commercial Website featuring the application for download for personal use – The family members of the subject may download the application for improving the subject’s general knowledge and current affairs happening around the world. This will make education more engaging for the subjects and they can eventually take part in contributing to a nation’s progress.

2. DVDs distributed among blind schools or government schools having access to computers – DVDs containing the copy of the software can be distributed among various government and blind schools where subjects can spend an hour with the application exploring the current trends across the world and giving interactive tests and receiving a score based performance which can be reflected in their academic reports.

3. The application can be used to conduct Joint Entrance Examinations even for the subjects and get them enrolled into reputed institutions based on their merit. Thus, ensuring no talent is left out in their journey towards excellence.

4. Android application can be downloaded on the subject’s mobile device and can be used to detect nearby objects with ease (feature under development).

Advantages and Disadvantages showing the effectiveness and limitations of the work

Advantages:

1. There are many advantages to the work that involves providing basic elementary education for the subjects to improving their knowledge about current affairs. Also, the application can be used to conduct Joint Entrance Examination since the application uses Firebase Authentication to authenticate users (which is a Google Auth tool) so it can provide the necessary security of the platform.

2. The application is the first of its kind application that practically involves no major hardware other than a camera. The platform can easily be replaced by a micro computer or anything that runs JAVA and Python natively (a microcontroller). Hence, the application is cross platform.

3. The application does not involve any sophisticated movements or wearable equipments on the subject so this makes the application user friendly for its end users.

Disadvantages:

1. The application requires proper lighting around the background where the camera will be set to work. Also, the light source must have an even distribution of intensity around the area and not fluctuating.

2. The application requires internet connectivity to gather current affairs updates and use cloud authentication services.

Future Scope

We are pleased to present the final native application which is fully functional with the demonstrated features. We are looking forward to optimize performance of our application and reduce CPU load for mobile platform. We have however, started working on an android application that helps the subjects determine nearby objects using voice responses. We are finding opportunities to collaborate with potential blind schools and hospitals to gain valuable feedback for developing a more sophisticated version of our application.

Conclusion

The application is the first of its kind application to help the subjects and also help the nation find new idols of talent to help with the nation’s progress. Hence, the name “Jyoti” which signifies hopes of education and progress not only in the mind of the subjects but also for the nation. The idea of the application evolved from the urgent need of bringing the Indian population victimized by visual impairment, dumbness or both to the main stream development and acquaints them with the new technologies that can change their lives for a better experience.

References:

[1] https://www.udemy.com/video-analytics-using-opencv-and-python-shells/
[2] https://github.com/Biswajee/TeamSpeak/
[3] https://github.com/Biswajee/ServerAnalyser/
[4] https://github.com/Biswajee/Sirena/
[5] http://sun.aei.polsl.pl/~mkawulok/gestures/
[6] http://image-net.org/
[7] https://github.com/tensorflow/models/tree/master/research/inception