

Big Data Analytical project for Climate Change Awareness

Team Name : Zeristos
Team Member : Naravula Loganathan, Barath - 28
Natesan Arumugam, Bharath Kumar - 29
Ramesh, Sibi Chakravarthy - 34

Framework Specification

Objectives

- **Project Objectives:**

The main objective of project is to create awareness for the climate change impact, presenting adaptation solution. This project provides science-based environmental education resource with credible information in form of video and image with support of assistant which is trained to answer question related to environment and climate change, thereby creating in-depth experience in virtual reality for user.

- **Project Motivation:**

The climate-change awareness that have emerged in the wake of massive fossil-fuel based industrialization indicate the need for a transition to sustainable energy, but attempts to create awareness and encourage people to follow pro-environmental behavior often have been limited and narrow reach of people and achieve only limited success. This problem of limited success is due to lack of awareness in people about its risk and danger that directly connected to climate change to them, upcoming generation and ecosystem.

This motivated us as create a big data analytics cum VR project for climate change awareness by bring all facts and its dangers effect on ecosystem to their own reality , so that they can feel it as if they are standing in melting glacier in Greenland, Sea-level rise or Extreme drought land in Africa and list is long.

"This is not just Academic project, but The project to save our Earth"

- **Significance / Uniqueness :**

1. Bringing Real-world Reality to our Virtual Reality
2. Interactive video presenting well-categorized section of Evidence, Cause, Effects, Scientific consequence, Vital Sign also with Remedies and Solution
3. Presenting Facts from well-published source in 360 video and Images where people can't go physically to create in-depth experience of facts.
4. Built-in Assistant to answer question related to climate change

- **Features: Use Case/Scenario**

Presenting data from well-published source in video and images where image and video are annotated. These annotated images are used to summarize into meaningful information , which is given to user via google home. In future increment we will present in the VR

Approach

- **Data Sources**

Data Set - Global Warming data

Data Category - 4 category in Global Warming as Below

1. Deforestation
2. Soil Acidification
3. Glacier Melting
4. Sea Level Rise

- **Analytic Tools**

The analytic tool used is the increment one of the project is **TensorFlow – Retrain Inception Final Layer**

- **Analytical Tasks**

1. This we take dataset from ImageNet (<http://www.image-net.org/>) and calculated training cost using TensorFlow.
2. Created Google conversation which interact with image analysis and image annotation.

- **Expected: Inputs/Outputs**

Input: Video (mkv format)

Output: Speech conversation using Google API

- **Algorithms**

Retrain Inception Final Layer is the algorithm used to train, test and predicate model for our climate change dataset.

Related Work

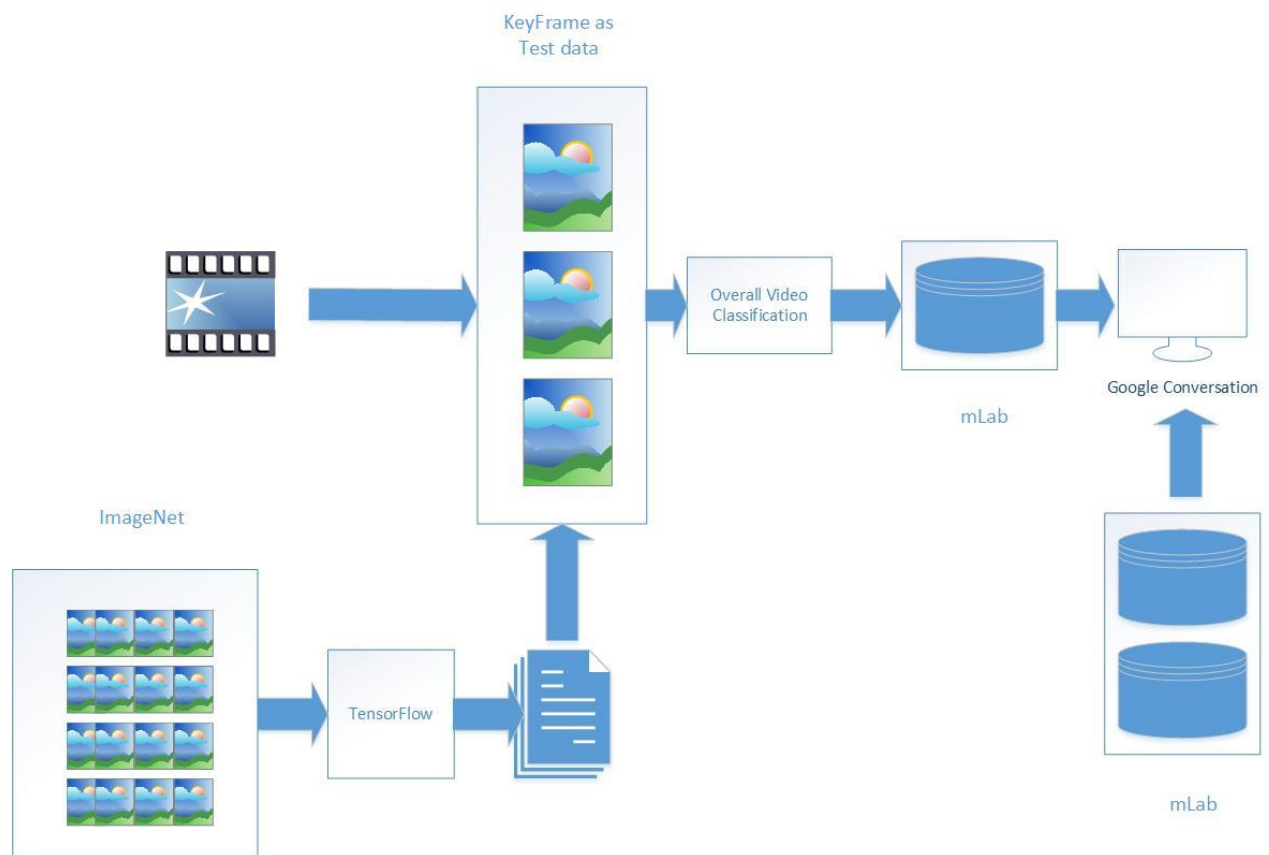
- **Open Source Projects**

The Scientists Using VR to Tackle Climate Change

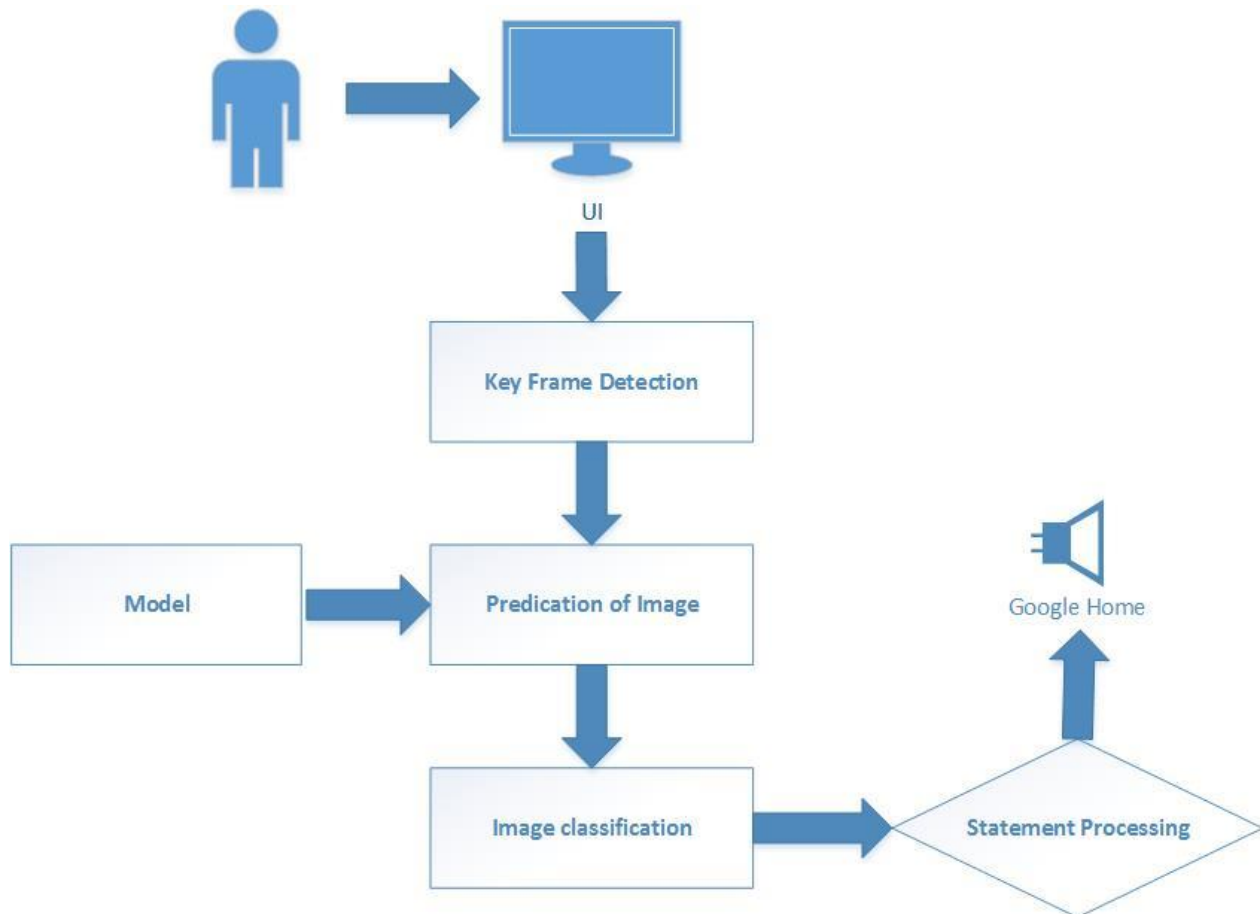
<http://thecreatorsproject.vice.com/blog/climate-change-vr-scientists>

Application Specification

Software Architecture



Activity Diagram



System Features

- User interface supported with video player.
- Detailed description about climate changes based on user preference
- Interactive user interface for user convenience
- Voice assistant support (By training data on climate change facts and information)

Existing Application/Service Used:

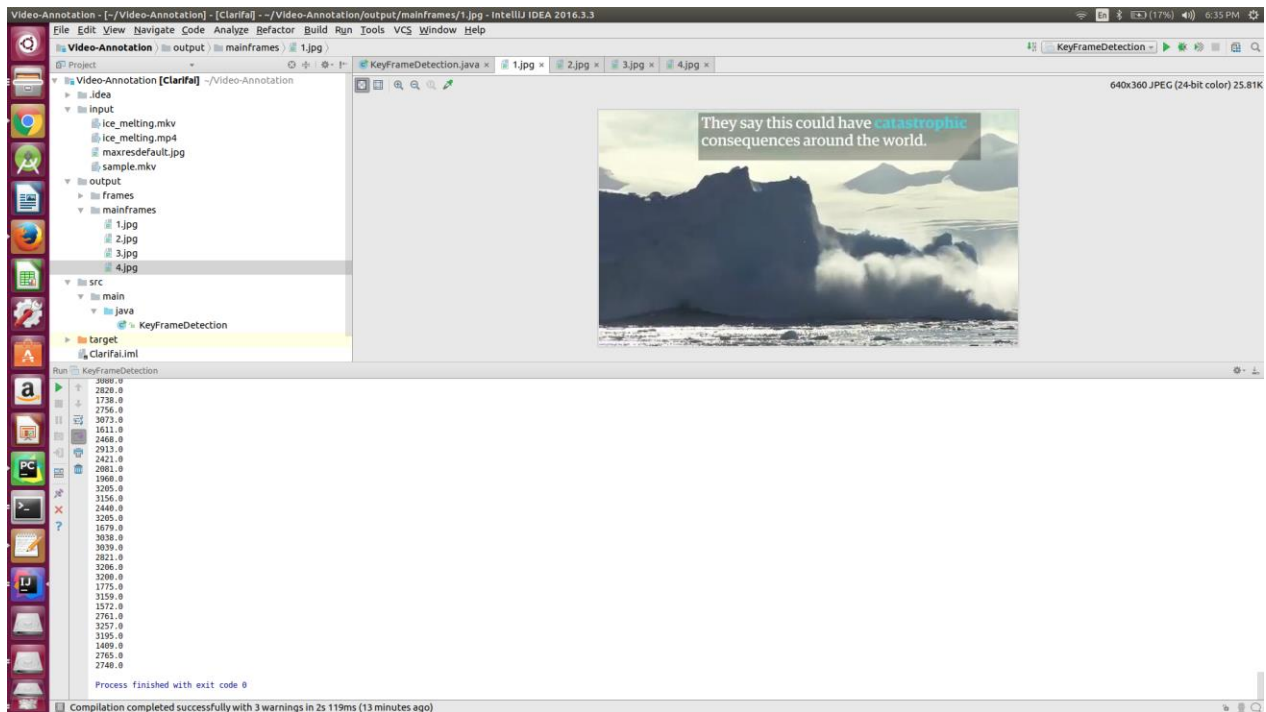
- Service Name: OpenIMAG
- Service Description: OpenIMAJ is set of libraries and tools used to extract main keyframes from video.
- Service URL: <http://openimaj.org/>

Implementation

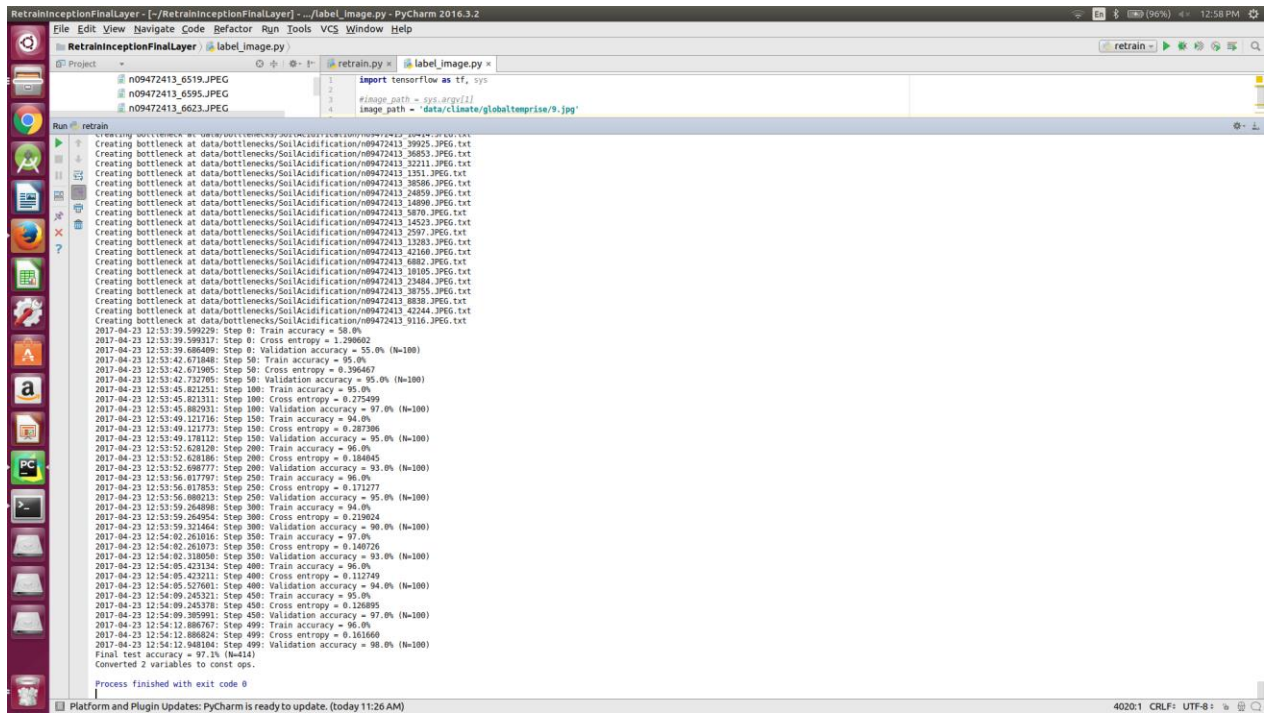
- First, we take climate change dataset from Image net and we train our model using Retrain Inception Final Layer model in Tensor Flow.
- Then we upload video in user interface and we extract mainframes from using image annotation tools such as Open IMAG.
- These collected mainframes are passed as test data to our trained model and image classification using softmax regression and the results are stored in Mlab database.
- For user convenience, we implemented Google Conversation API using api.ai and created intents according to our theme.
- In simulation when these intents are triggered we fetch the desired results from mlab database which is deployed in Heroku-cloud application

Documentation:

Video Annotation



Tensor Flow – Training

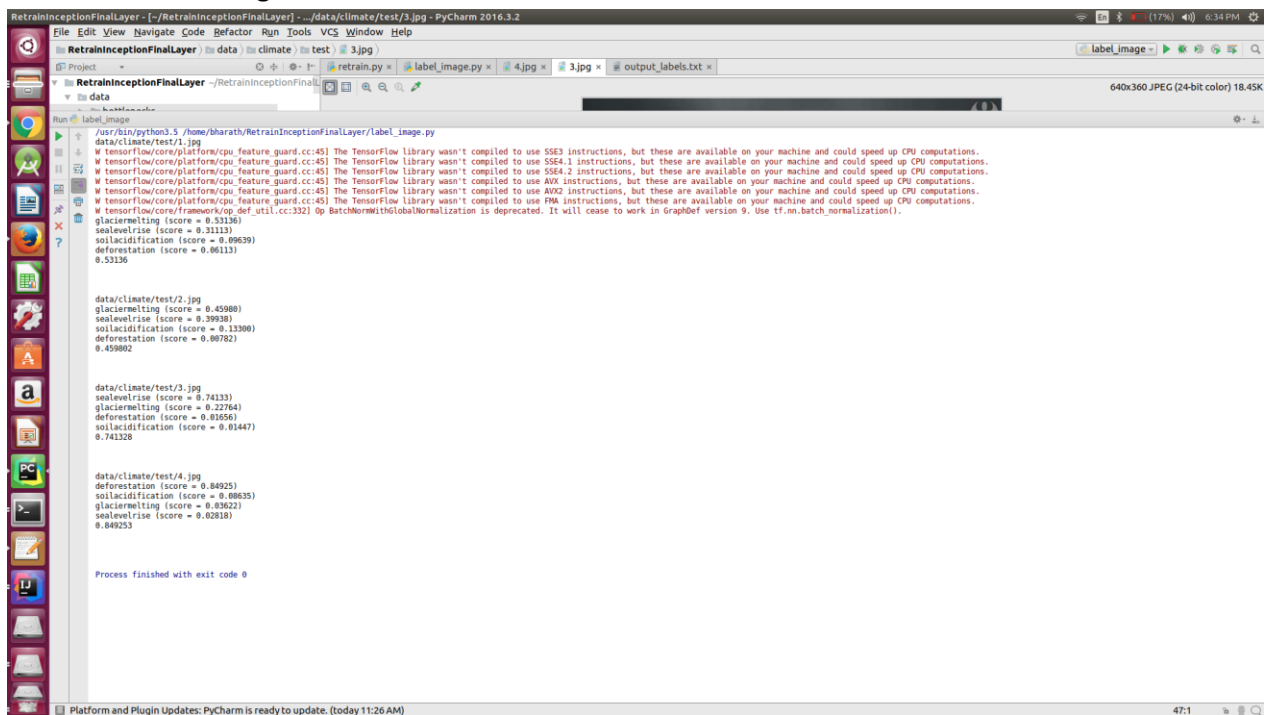


The screenshot shows the PyCharm IDE with the 'RetrainInceptionFinalLayer' project. The 'Run' console displays the output of the 'retrain.py' script. The script imports TensorFlow and sets the image path to 'data/climate/globaltempire/9.jpg'. The output shows the creation of bottleneck files for various images and the training progress. The training process includes steps for training accuracy, cross entropy, and validation accuracy. The final test accuracy is 97.1% (N=141).

```
RetrainInceptionFinalLayer - [RetrainInceptionFinalLayer] - label_image.py - PyCharm 2016.3.2
RetrainInceptionFinalLayer - [RetrainInceptionFinalLayer] - label_image.py
retrain.py
import tensorflow as tf, sys
image_path = sys.argv[1]
image_path = 'data/climate/globaltempire/9.jpg'

Creating bottleneck at data/bottlenecks/SoilAcidification/n9472413_39925.JPEG.txt
Creating bottleneck at data/bottlenecks/SoilAcidification/n9472413_36853.JPEG.txt
Creating bottleneck at data/bottlenecks/SoilAcidification/n9472413_32211.JPEG.txt
Creating bottleneck at data/bottlenecks/SoilAcidification/n9472413_1351.JPEG.txt
Creating bottleneck at data/bottlenecks/SoilAcidification/n9472413_38586.JPEG.txt
Creating bottleneck at data/bottlenecks/SoilAcidification/n9472413_24859.JPEG.txt
Creating bottleneck at data/bottlenecks/SoilAcidification/n9472413_14808.JPEG.txt
Creating bottleneck at data/bottlenecks/SoilAcidification/n9472413_5870.JPEG.txt
Creating bottleneck at data/bottlenecks/SoilAcidification/n9472413_14523.JPEG.txt
Creating bottleneck at data/bottlenecks/SoilAcidification/n9472413_2597.JPEG.txt
Creating bottleneck at data/bottlenecks/SoilAcidification/n9472413_13283.JPEG.txt
Creating bottleneck at data/bottlenecks/SoilAcidification/n9472413_42160.JPEG.txt
Creating bottleneck at data/bottlenecks/SoilAcidification/n9472413_6882.JPEG.txt
Creating bottleneck at data/bottlenecks/SoilAcidification/n9472413_10105.JPEG.txt
Creating bottleneck at data/bottlenecks/SoilAcidification/n9472413_23464.JPEG.txt
Creating bottleneck at data/bottlenecks/SoilAcidification/n9472413_38755.JPEG.txt
Creating bottleneck at data/bottlenecks/SoilAcidification/n9472413_8838.JPEG.txt
Creating bottleneck at data/bottlenecks/SoilAcidification/n9472413_42244.JPEG.txt
Creating bottleneck at data/bottlenecks/SoilAcidification/n9472413_9116.JPEG.txt
2017-04-23 12:53:39.596229: Step 0: Train accuracy = 58.0%
2017-04-23 12:53:39.599317: Step 0: Cross entropy = 1.298602
2017-04-23 12:53:39.606409: Step 0: Validation accuracy = 55.0% (N=100)
2017-04-23 12:53:42.671848: Step 50: Train accuracy = 95.0%
2017-04-23 12:53:42.671985: Step 50: Cross entropy = 0.396467
2017-04-23 12:53:42.732705: Step 50: Validation accuracy = 95.0% (N=100)
2017-04-23 12:53:45.821251: Step 100: Train accuracy = 95.0%
2017-04-23 12:53:45.821311: Step 100: Cross entropy = 0.275497
2017-04-23 12:53:45.882931: Step 100: Validation accuracy = 97.0% (N=100)
2017-04-23 12:53:49.121716: Step 150: Train accuracy = 94.0%
2017-04-23 12:53:49.121773: Step 150: Cross entropy = 0.287386
2017-04-23 12:53:49.178112: Step 150: Validation accuracy = 95.0% (N=100)
2017-04-23 12:53:52.628128: Step 200: Train accuracy = 96.0%
2017-04-23 12:53:52.628186: Step 200: Cross entropy = 0.384845
2017-04-23 12:53:52.688777: Step 200: Validation accuracy = 93.0% (N=100)
2017-04-23 12:53:56.017797: Step 250: Train accuracy = 96.0%
2017-04-23 12:53:56.017853: Step 250: Cross entropy = 0.371277
2017-04-23 12:53:56.088213: Step 250: Validation accuracy = 95.0% (N=100)
2017-04-23 12:53:59.264808: Step 300: Train accuracy = 94.0%
2017-04-23 12:53:59.264954: Step 300: Cross entropy = 0.219024
2017-04-23 12:53:59.321464: Step 300: Validation accuracy = 96.0% (N=100)
2017-04-23 12:54:02.261016: Step 350: Train accuracy = 97.0%
2017-04-23 12:54:02.261073: Step 350: Cross entropy = 0.148726
2017-04-23 12:54:02.318958: Step 350: Validation accuracy = 95.0% (N=100)
2017-04-23 12:54:05.423134: Step 400: Train accuracy = 96.0%
2017-04-23 12:54:05.423211: Step 400: Cross entropy = 0.112748
2017-04-23 12:54:05.527001: Step 400: Validation accuracy = 94.0% (N=100)
2017-04-23 12:54:09.245321: Step 450: Train accuracy = 95.0%
2017-04-23 12:54:09.245378: Step 450: Cross entropy = 0.126895
2017-04-23 12:54:09.385991: Step 450: Validation accuracy = 97.0% (N=100)
2017-04-23 12:54:12.886767: Step 499: Train accuracy = 96.0%
2017-04-23 12:54:12.886824: Step 499: Cross entropy = 0.161660
2017-04-23 12:54:12.948184: Step 499: Validation accuracy = 98.0% (N=100)
Final test accuracy = 97.1% (N=141)
Converted 2 variables to const ops.
Process finished with exit code 0
```

Tensor Flow – Testing



The screenshot shows the PyCharm IDE with the 'RetrainInceptionFinalLayer' project. The 'Run' console displays the output of the 'test.py' script. The script imports TensorFlow and sets the image path to 'data/climate/test/1.jpg'. The output shows the testing results for various images, including 'data/climate/test/1.jpg', 'data/climate/test/2.jpg', 'data/climate/test/3.jpg', and 'data/climate/test/4.jpg'. The testing process includes steps for training accuracy, cross entropy, and validation accuracy. The final test accuracy is 97.1% (N=141).

```
RetrainInceptionFinalLayer - [RetrainInceptionFinalLayer] - data/climate/test/1.jpg - PyCharm 2016.3.2
RetrainInceptionFinalLayer - [RetrainInceptionFinalLayer] - data/climate/test/1.jpg
data/climate/test/1.jpg
/usr/bin/python3.5 /home/bharath/RetrainInceptionFinalLayer/label_image.py
W tensorflow/core/platform/cpu_feature_guard.cc:45 The TensorFlow Library wasn't compiled to use SSE3 instructions, but these are available on your machine and could speed up CPU computations.
W tensorflow/core/platform/cpu_feature_guard.cc:45 The TensorFlow Library wasn't compiled to use SSE4.1 instructions, but these are available on your machine and could speed up CPU computations.
W tensorflow/core/platform/cpu_feature_guard.cc:45 The TensorFlow Library wasn't compiled to use SSE4.2 instructions, but these are available on your machine and could speed up CPU computations.
W tensorflow/core/platform/cpu_feature_guard.cc:45 The TensorFlow Library wasn't compiled to use AVX instructions, but these are available on your machine and could speed up CPU computations.
W tensorflow/core/platform/cpu_feature_guard.cc:45 The TensorFlow Library wasn't compiled to use AVX2 instructions, but these are available on your machine and could speed up CPU computations.
W tensorflow/core/platform/cpu_feature_guard.cc:45 The TensorFlow Library wasn't compiled to use FMA instructions, but these are available on your machine and could speed up CPU computations.
glacermelting (score = 0.53136)
sealevelrise (score = 0.31113)
soilacidification (score = 0.69639)
deforestation (score = 0.06113)
0.53136

data/climate/test/2.jpg
glacermelting (score = 0.45986)
sealevelrise (score = 0.39598)
soilacidification (score = 0.13300)
deforestation (score = 0.80782)
0.45986

data/climate/test/3.jpg
sealevelrise (score = 0.74133)
glacermelting (score = 0.22764)
deforestation (score = 0.80506)
soilacidification (score = 0.81447)
0.74133

data/climate/test/4.jpg
deforestation (score = 0.84925)
soilacidification (score = 0.88635)
glacermelting (score = 0.83622)
sealevelrise (score = 0.62618)
0.84925

Process finished with exit code 0
```

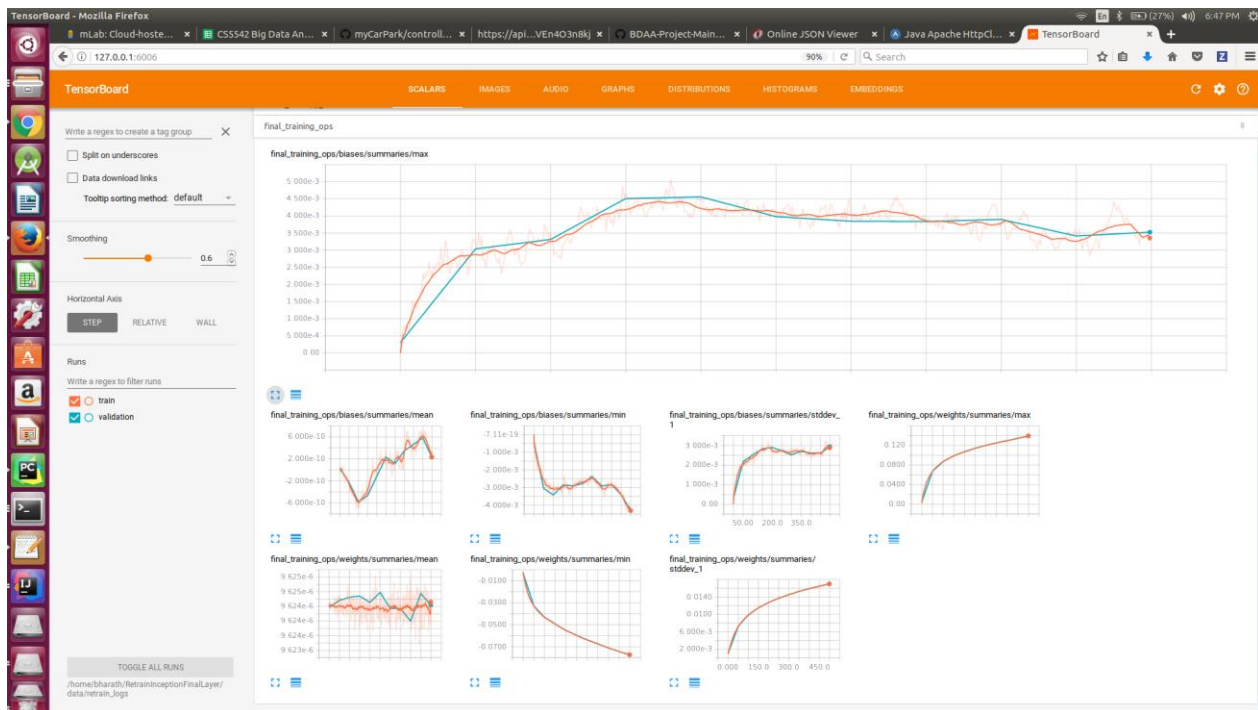
Database

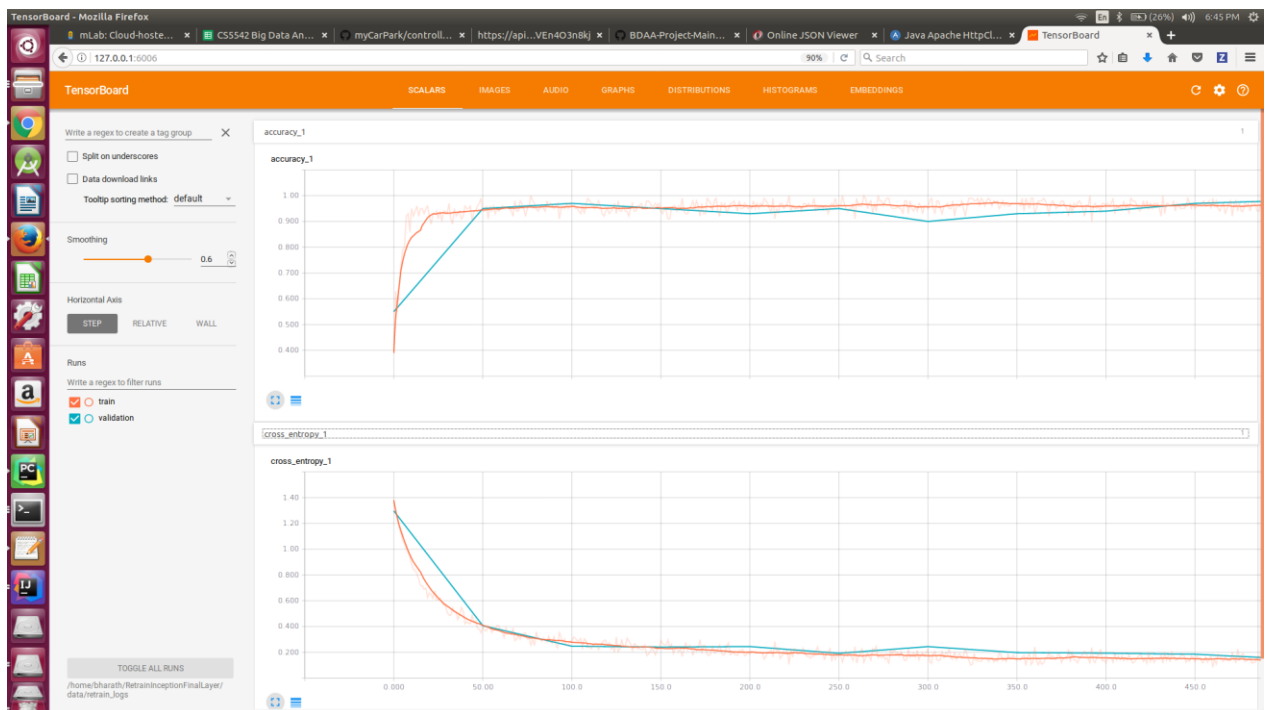
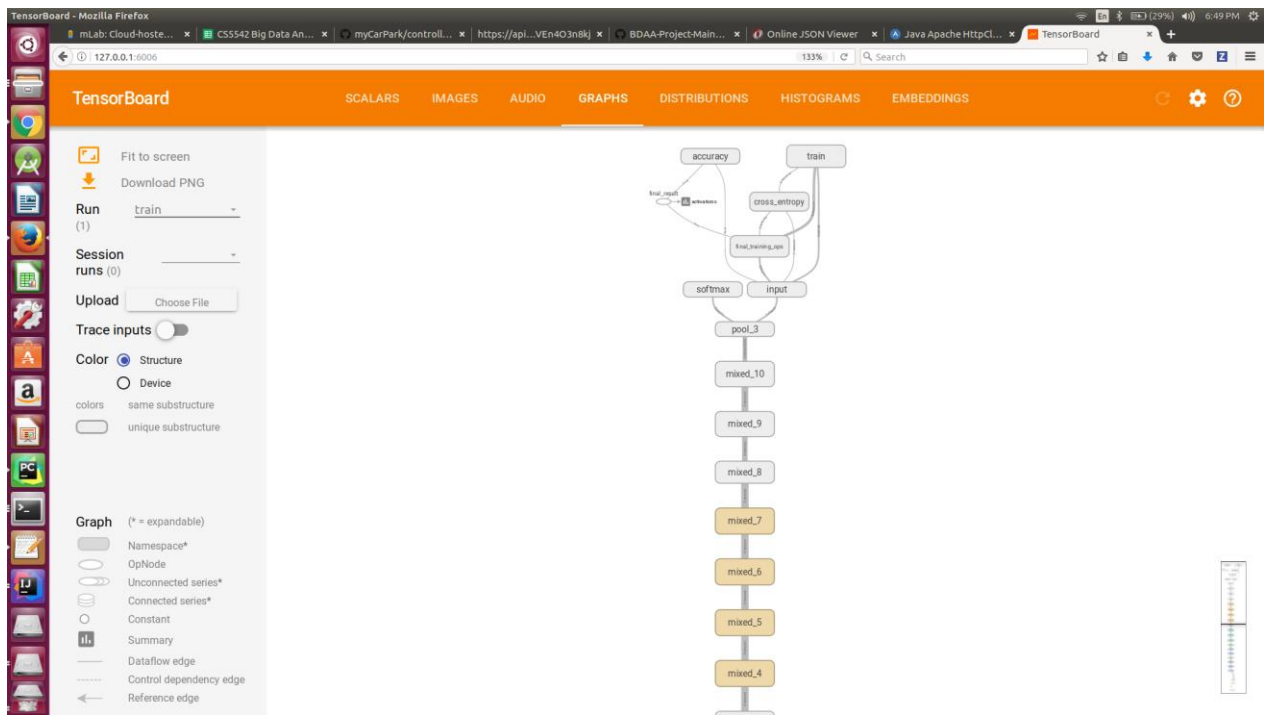
The screenshot shows the mLab MongoDB web interface in a Mozilla Firefox browser. The address bar displays the URL `https://mlab.com/databases/api/database/collections/test`. The interface includes a search bar at the top with the text "Start new search". Below the search bar, the "All Documents" section is visible, showing a list of documents in a table view. The documents are displayed in a dark theme with a light blue border. Each document contains a unique `_id`, a `name`, and a `value`. The documents are as follows:

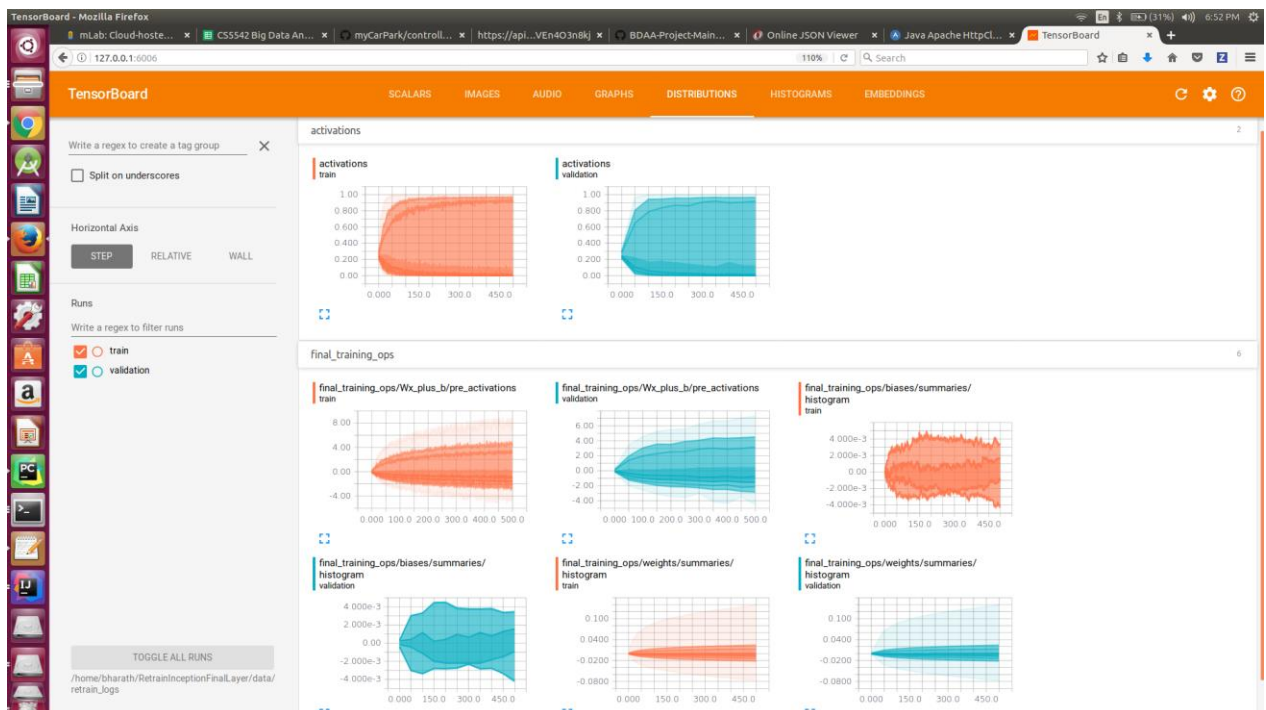
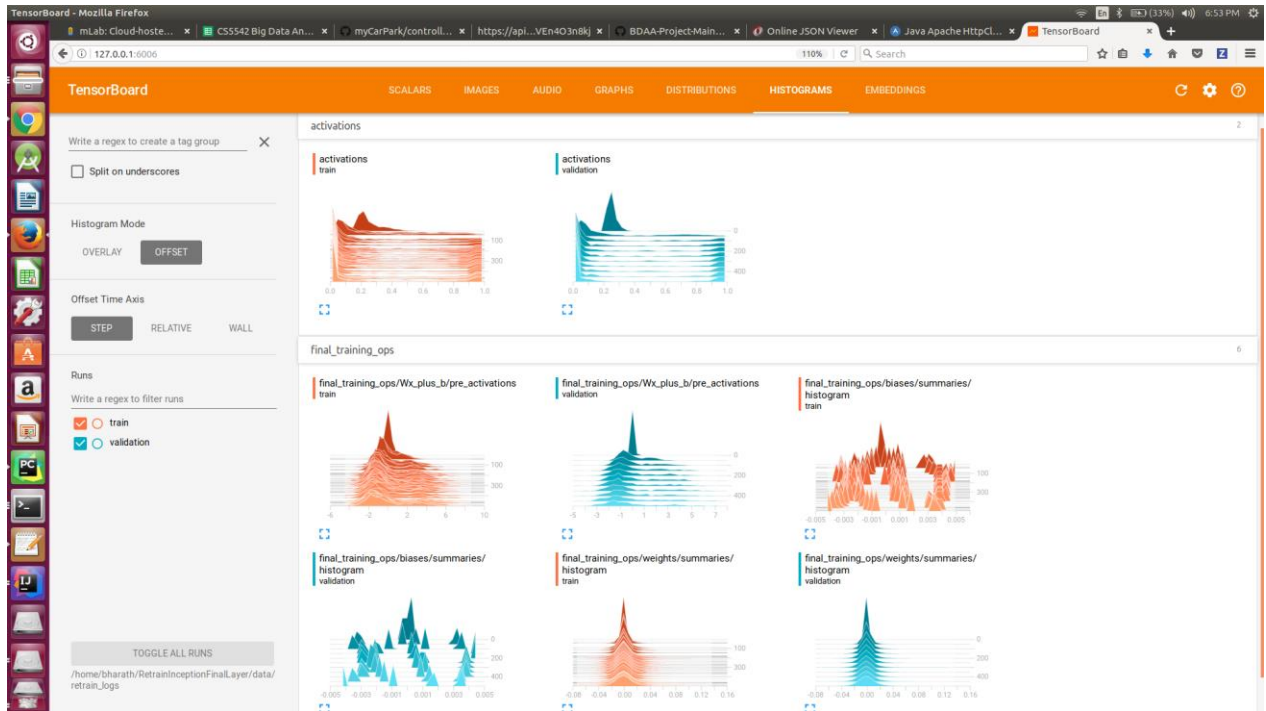
_id	name	value
58f39cd8bc2838932fb6c	glaciemelting	0.53136
58f39fd8bc2838932fb6e	glaciemelting	0.459802
58f3a0d8bc2838932fb70	sealevelrise	0.741328
58f3a0d8bc2838932fb72	deforestation	0.849253

At the bottom of the document list, there is a "records / page" dropdown menu set to 10, and a "[1 - 4 of 4]" indicator. A sidebar on the left contains various application icons, and a top navigation bar includes links to "mLab Cloud-hosted MongoDB", "CS5542 Big Data An...", "myCarPark/controll...", "https://api...VEn403n8kj", "BDAA-Project-Main...", "Online JSON Viewer", and "Java Apache HttpCl...".

Tensor Board







Google Conversation – Heroku Deployment

```
bharath@bharath-Inspiron-13-7359: ~/Google-Conversation/ClientApp/target$ heroku login
Enter your Heroku credentials.
Email: bn4n5@mail.unkc.edu
Password (typing will be hidden):
Logged in as bn4n5@mail.unkc.edu
bharath@bharath-Inspiron-13-7359:~/Google-Conversation/ClientApp/target$ heroku deploy:war Client.war --app projectbigdata
Uploading Client.war
-----> Packaging application...
- app: projectbigdata
- including: webapp-runner.jar
- including: Client.war
-----> Creating build...
- file: slug.tgz
- size: 24MB
-----> Uploading build...
- success
-----> Deploying...
remote: -----> heroku-deploy app detected
remote: -----> Installing OpenJDK 1.8... done
remote: -----> Discovering process types
remote: Procfile declares types => web
remote: -----> Compressing...
remote: Done: 72.9%
remote: -----> Launching...
remote: Released v6
remote: https://projectbigdata.herokuapp.com/ deployed to Heroku
-----> Done
bharath@bharath-Inspiron-13-7359:~/Google-Conversation/ClientApp/target$
```

Google Conversation – Api.ai

The screenshot displays the API.AI console interface. On the left, a sidebar contains navigation options: Intents, Entities, Training, Integrations, Fulfillment, Prebuilt Agents, Small Talk, Docs, and Forum. The main area is titled 'climate' and shows a list of user intents with their corresponding responses. The 'User says' section lists four intents: 'what is the solution of this climate change?', 'what is the effect of this climate change?', 'what is the cause of this climate change?', and 'what is the summary of this climate change?'. The 'Response' section shows the corresponding AI responses for each intent. The 'Entities' section lists the entity 'climate' with its value 'solution'.

PARAMETER NAME	ENTITY	RESOLVED VALUE
climate	@climate	solution

The 'Response' section shows the AI's response for each intent. The first response is 'Efforts to stop or slow deforestation can be carried out by transfer land rights from public to its indigenous inhabitant. Also reduction in emission to limit and/or roll back deforestation.' The second response is 'Not available'. The third response is 'Not available'. The fourth response is 'Not available'.

API.AI - Google Chrome

climate

climate_change

Intents

Entities

Training

Integrations

Fulfillment

Prebuilt Agents

Small Talk

Docs

Forum

Account

Logout

Define synonyms ☒ Allow automated expansion

cause	reason, cause, source, root, origin
effect	effects, consequences, outcome, result
solution	solution, remedies, prevention
summary	summary, abstract, synopsis, outline, review

Click here to edit entry

+ Add a row

Agent

USER SAYS [COPY CURL](#)

what is the solution of this climate change?

RESPONSE [PLAY](#)

Efforts to stop or slow deforestation can be carried out by transfer land rights from public to its indigenous inhabitant. Also reduction in emission to limit and/or roll back deforestation.

INTENT

climate

ACTION

Not available

PARAMETER	VALUE
climate	solution

SHOW JSON

Svetlana from API.AI

Hey Bharath, Welcome to API.AI! Here are a few resources to get you started...

Google Conversation – Webhook

API.AI - Google Chrome

Fulfillment

climate_change

Intents

Entities

Training

Integrations

Fulfillment

Prebuilt Agents

Small Talk

Docs

Forum

Account

Logout

Webhook

Your web service will receive a POST request from API.AI in the form of the response to a user query matched by intents with webhook enabled. Be sure that your web service meets all the [webhook requirements](#).

[Webhook example](#)

ENABLED ☒

URL*

<https://projectbigdata.herokuapp.com/webhook>

BASIC AUTH

Enter username...

Enter password...

HEADERS

Enter key...

Enter value...

Enter key...

Enter value...

[Add header](#)

DOMAINS

Disable webhook for all domains

Agent

USER SAYS [COPY CURL](#)

what is the solution of this climate change?

RESPONSE [PLAY](#)

Efforts to stop or slow deforestation can be carried out by transfer land rights from public to its indigenous inhabitant. Also reduction in emission to limit and/or roll back deforestation.

INTENT

climate

ACTION

Not available

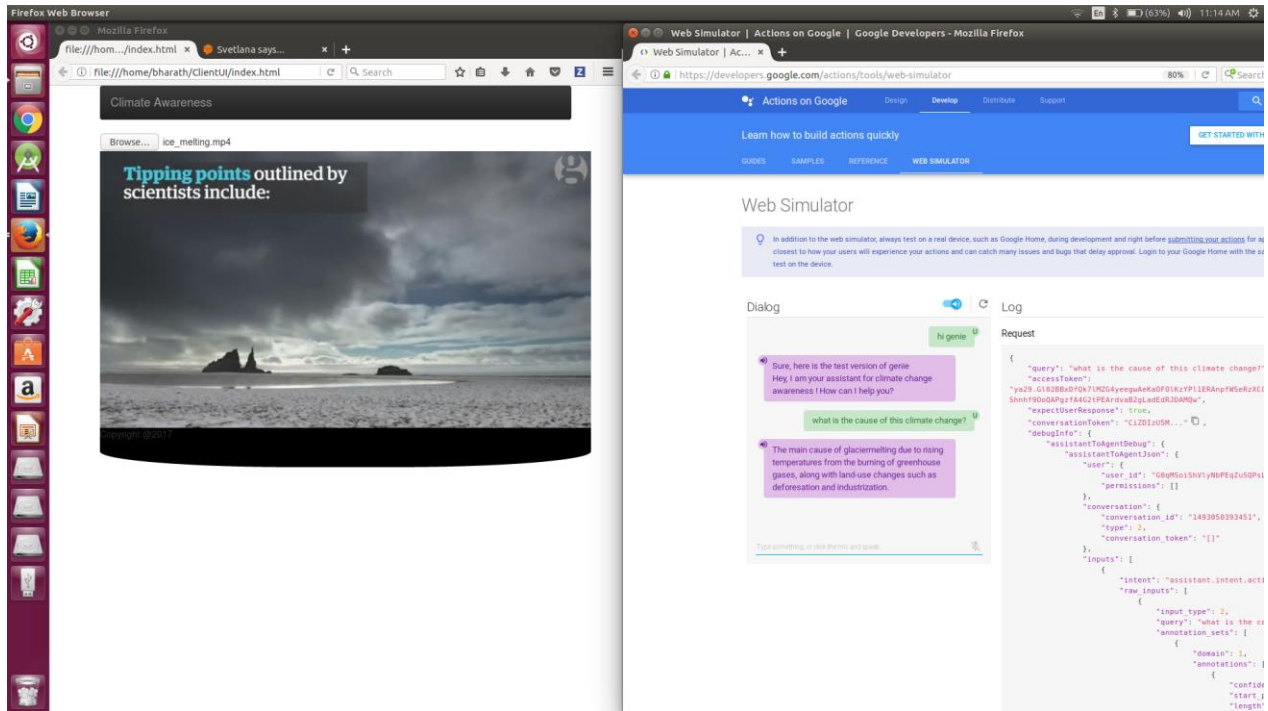
PARAMETER	VALUE
climate	solution

SHOW JSON

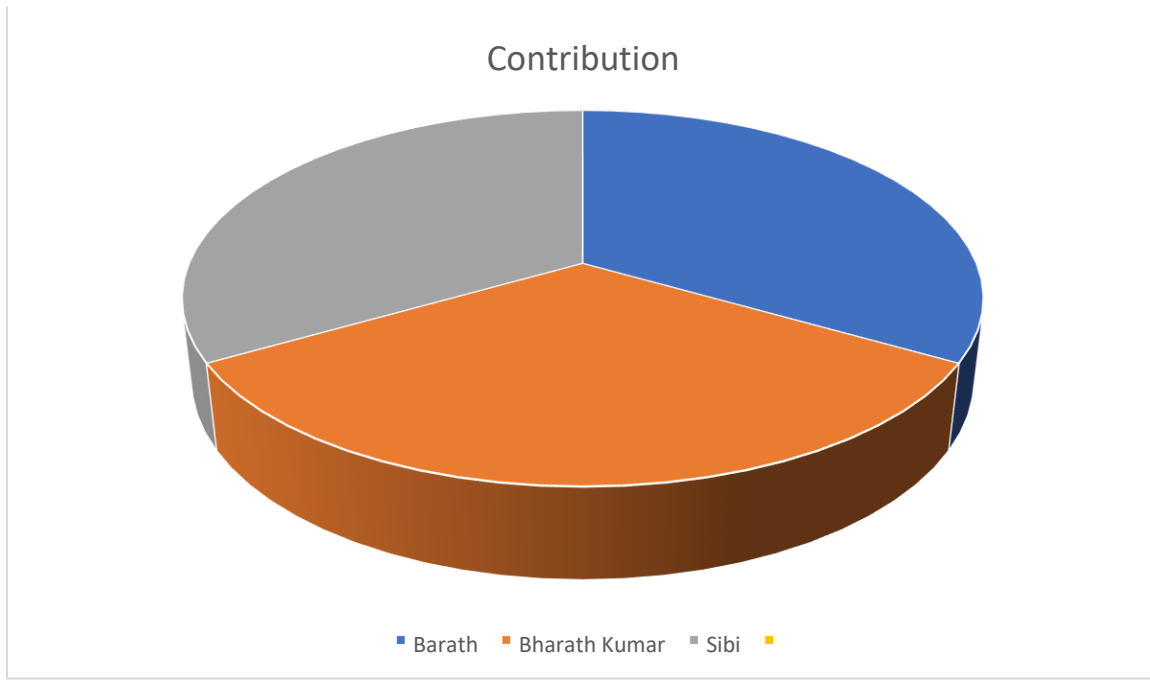
Svetlana from API.AI

Hey Bharath, Welcome to API.AI! Here are a few resources to get you started...

Home Page



Project Management



Contribution in Project:

Naravula Loganathan,Barath – 28

- Documentation
- Google Conversation API

Natesan Arumugam,Bharath Kumar – 29

- Tensor Flow Training Model
- Database Schema – Mlab

Ramesh,Sibi Chakravarthy – 34

- User Interface
- Key Frame Detection