UMKC

UNIVERSITY OF MISSOURI
KANSAS CITY

Mood Detection and Management Using Facial Expression

Fatema Hasta | 16247903

Satya Venkata Ranga Janaki Sriram Mentey | 16251319

Maniraj Mohareer | 16251433

Sunil Kumar Madikanti | 16254143

Project Objectives:

- The core idea of this project is to classify a person's mood using his facial expression and provide certain features according to the mood. The features include suggesting a song as per the mood, suggesting movies, near-by get away places and management tips for the particular mood. A web application can be designed that takes the input in the form of an image. This image can be fed to our model that detects the emotion and classifies the image either into happy class, sad class, anger class or suprise class. Based on this class, particular features will be appeared to the user. The moods of the person can be tracked everyday and an analysis chart could be given at the end of particular month.

1) Analyze the mood of the person with the facial expression.

The model can detect like four to five different moods of the person. The classes can be divided based on the emotion like happiness, sadness, excitement and anger. Based on the facial expression the model predicts the emotion that would be useful
further in the application.

2) Give some suggestions based on the current emotion like playing songs, suggesting get aways. On the user’s profile in the website, he/she can have access to a lot of
information that is sorted in detail. They include:

• Suggesting movies based on the current emotion of the user.
• Redirecting to music and play songs or suggest songs based on that particular mood.
• Suggesting him certain places that can change his/her mood .
• Emotion Management tips.

3) Provide a monthly analysis of the emotions to the user using charts and Graphs.

- System Features:
  - Image as an Input
    - We can upload the image as the input to the application. On analyzing the image, the emotions are predicted and the conclusion is omitted out.

- Most precise conclusion

- Approach:
  - Data Sources
    - We have taken images of faces from http://www.face-rec.org/databases/.
    - We have classified the dataset into 10 classes: ‘angry’, ‘contempt’, ‘disgust’, ‘excited’, ‘fear’, ‘happy’, ‘neutral’, ‘sad’, ‘scared’, ‘surprise’.
  - Analytical tools
    - Apache Spark
  - IDE
    - Intellij
  - Expected Inputs/Outputs
    - Input: Input image of facial expression of a person.
    - Output: The expression of the person and suggestions for Movies, Music, Getaway locations.
  - Algorithms
    - The algorithms used in our project is K-Means clustering and Random Forest classification.
    - K-means is used for clustering our feature vectors.
    - Random forest is used for test image classification.

- **Related Work**
  - We did a lot of ground work, there is no project of this kind so far. We want to make this completely functional and achieve all the objectives and make this one-of-a-kind.

Application Specification & Implementation

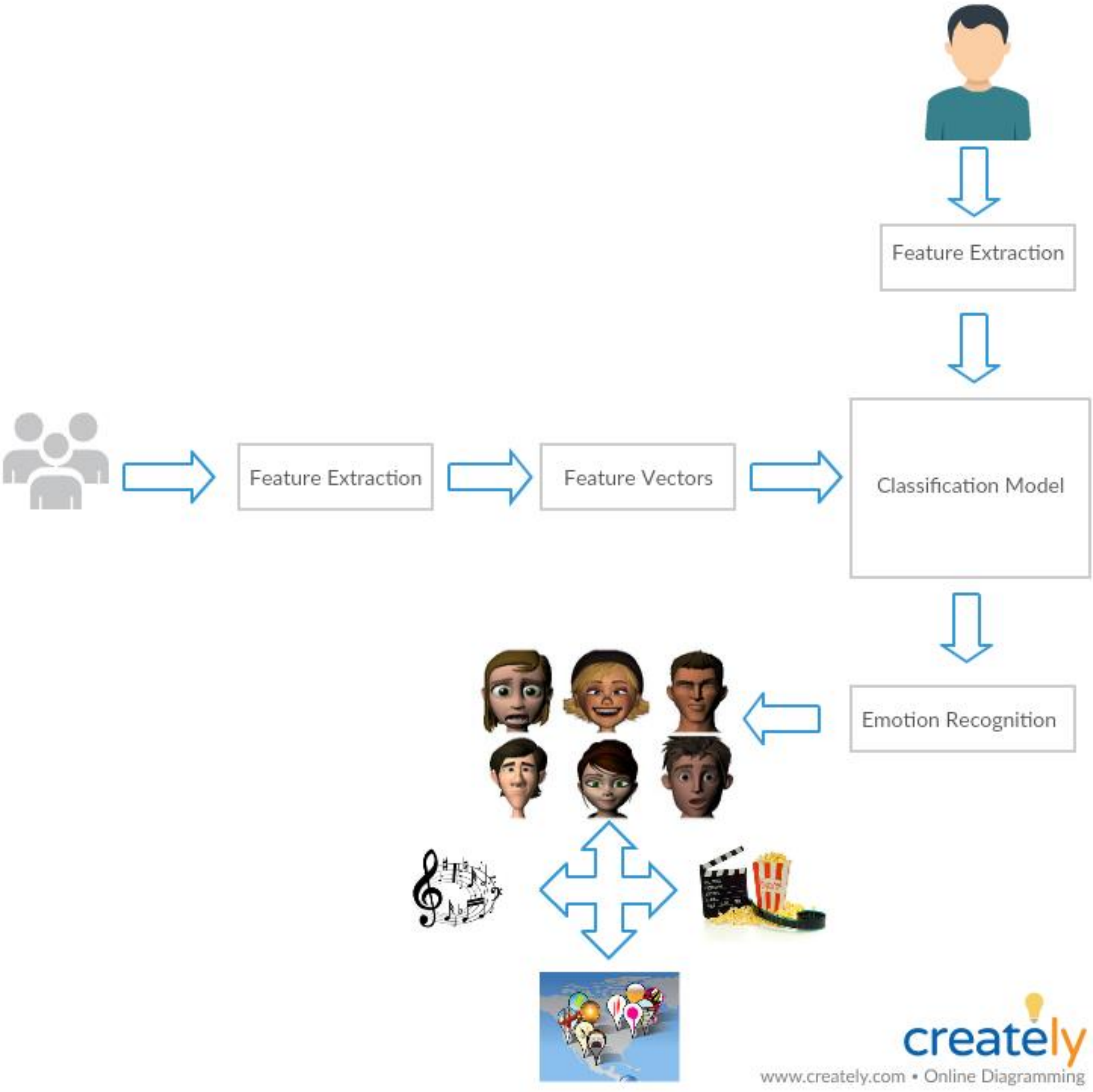
- **System Specification (Big Data Analytics Server/Client)**

MacOS Sierra+/ Windows 7 + Ubuntu

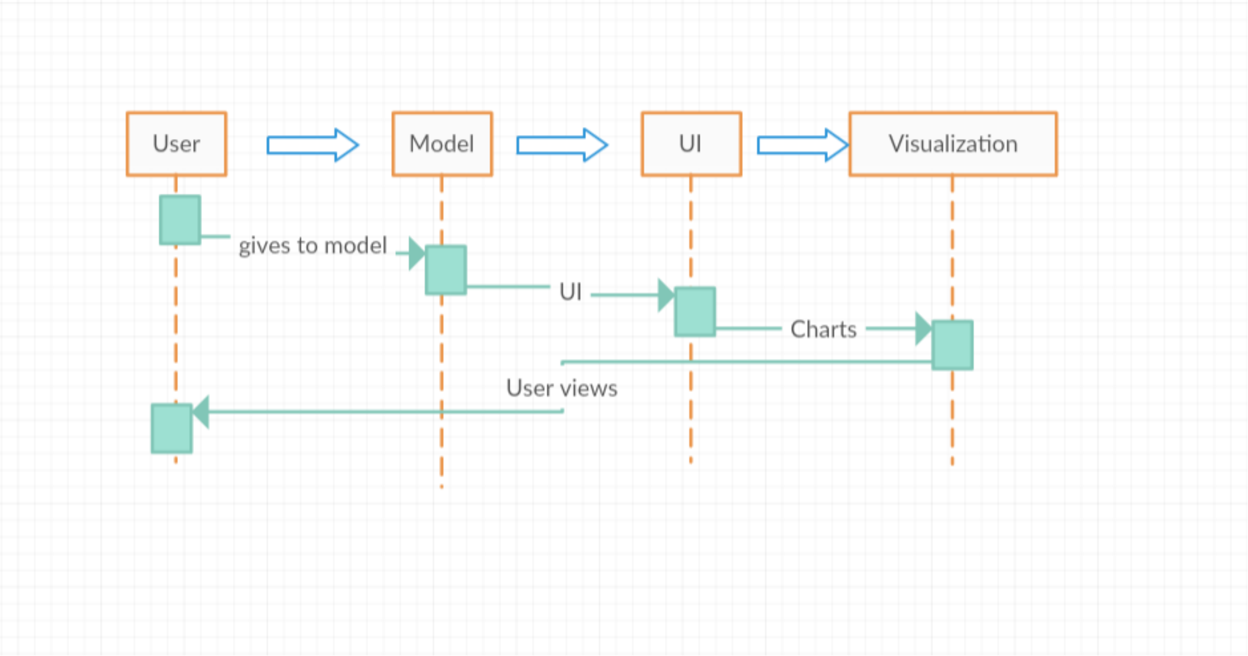
2.3 GHz Intel Core i5 and above

Minimum 8 GB 2133 MHz LPDDR3 and above

- **Software Architecture**



- **Sequence Diagram**



- **Features, workflow, technologies**

Face Detection and Feature Extraction

First, one frame in video stream is grabbed on the mobile device. In a frame, the face is found by facial detection module. Through ASM module implemented, 77 facial landmarks are located on a face, and then based on x, y coordinates of landmarks, 13 high-level facial shape features are generated and normalized. If this face has a neutral expression, the system keeps current features as neutral features which will be used as base features of calculating displacement between features later. Otherwise, if it is not a neutral expression, the system generates new features by calculating the displacement between current features and neutral features.

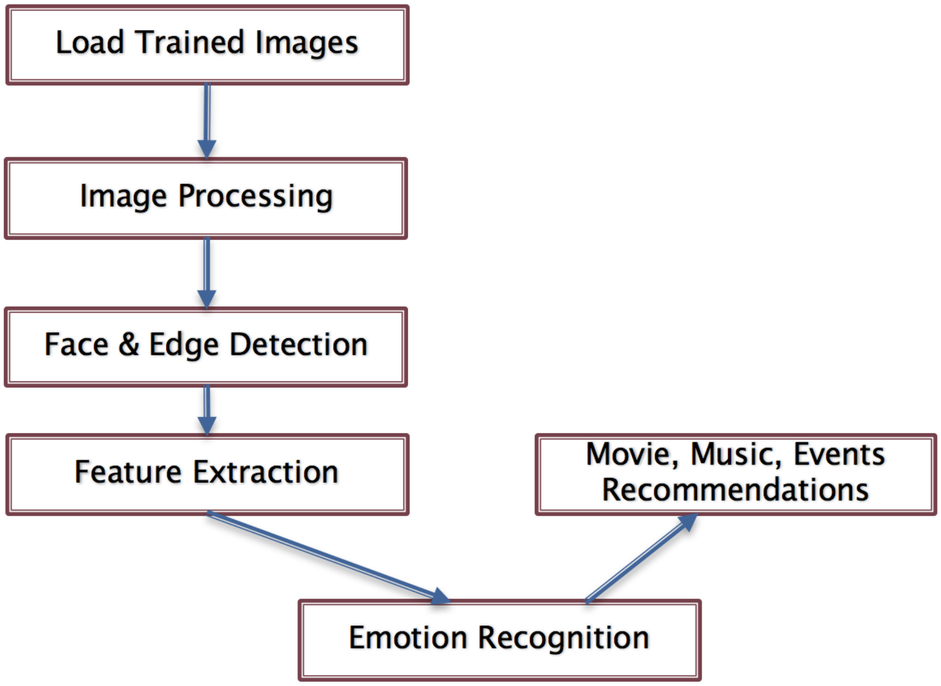
Related Work:

- We did a lot of ground work, there is no project of this kind so far. We want to make this completely functional and achieve all the objectives and make this one-of -a-kind.

Technologies

- Spark
- HTML5
- Angular
- CSS
- Bootstrap
- REST API
- Scala
- Azure (Planning to Use)

- **Activity Diagram (workflow, data, task)**



- **System Features:**
    - Image as an Input
        - We can upload the image as the input to the application. On analyzing the image, the emotions are predicted and the conclusion is omitted out.
    - Most precise conclusion

- **Approach:**
    - **Data Sources**
        - We have taken images of faces from http://www.face-rec.org/databases/.
        - We have classified the dataset into 10 classes: ‘angry’, ‘contempt’, ‘disgust’, ‘excited’, ‘fear’, ‘happy’, ‘neutral’, ‘sad’, ‘scared’, ‘surprise’.
    - **Analytical tools**
        - Apache Spark
    - **IDE**
        - Intellij

- **Expected Inputs/Outputs**
  - Input: Input image of facial expression of a person.
  - Output: The expression of the person and suggestions for Movies, Music, Getaway locations.
- **Algorithms**
  - The algorithms used in our project is K-Means clustering and Random Forest classification.
  - K-means is used for clustering our feature vectors.
  - Random forest is used for test image classification.

- **Existing Applications/Services Used:**

i. **Kairos** – Offers a wide variety of image recognition solutions through their API. Their API endpoints include identifying gender, age, emotional depth, facial recognition in both photo and video, and more.

ii. **Trueface.ai** – One flaw with some facial recognition APIs is that they are unable to differentiate between a face and a picture of a face. TrueFace.ai solves that problem with their ability to do spoof detection through their API.

iii. **Amazon Recognition** – This facial recognition API is fully integrated into the Amazon Web Service ecosystem. Using this API will make it really easy to build applications that make use of other AWS products.

iv. **Face Recognition and Face Detection by Lambda Labs** – With over 1,000 calls per month in the free pricing tier, and only $0.0024 per extra API call, this API is a really affordable option for developers wanting to use a facial recognition API.

v. **EmoVu by Eyeris** – This API was created by Eyeris and it is a deep learning-based emotion recognition API. EmoVu allows for great emotion recognition results by identifying facial micro-expressions in real-time.

vi. **Microsoft Face API** – One cool feature that I found while doing research on the Microsoft Face API, is that the API has the ability to do “similar face search.” When this API endpoint is given a collection of faces, and a new face as a query, the API will return a collection of similar faces from the collection.

vii. **Animetrics Face Recognition** – Using advanced 2D-to-3D algorithms, this API will convert a 2D image into a 3D model. The 3D model will then be used for facial recognition purposes.

viii. **Face++** – This API also has an offline SDK for iOS & Android for you to use. The offline SDK does not provide face recognition, but it can perform face detection, comparing, tracking and landmarks, all while the phone does not have cell service.

ix. **Google Cloud Vision** – By being integrated into the Google Cloud Platform, this API will be a breeze for you to integrate into applications that are already using other Google Cloud Platform products and services.

x. **IBM Watson Visual Recognition** – Whether it is faces, objects, colors, or food, this API lets you identify many different types of classifiers. If the included classifiers aren't enough, then you can train and use your own custom classifiers.

References:

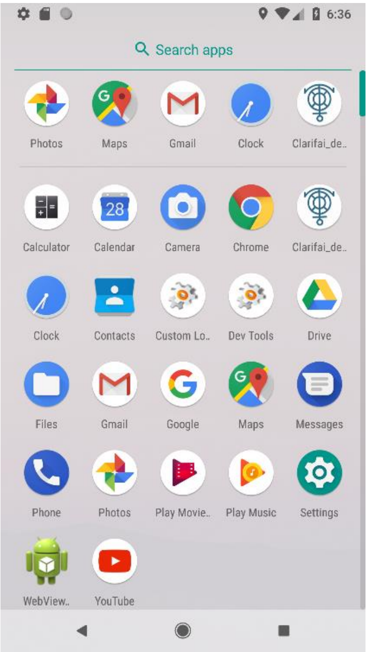
Article: Facial Expression Emotion Detection for Real-Time Embedded Systems †

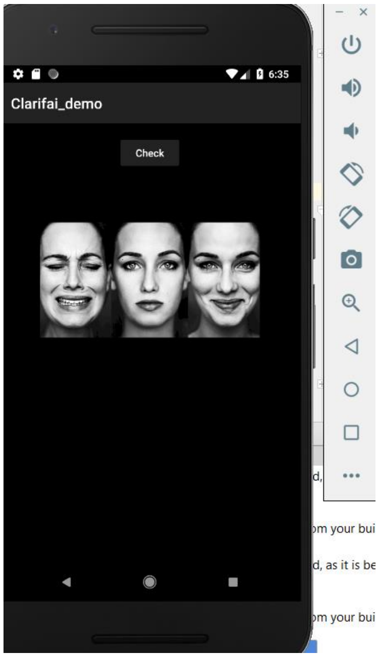
http://openaccess.thecvf.com/content_cvpr_workshops_2014/W03/papers/Suk_Real-time_Mobile_Facial_2014_CVPR_paper.pdf

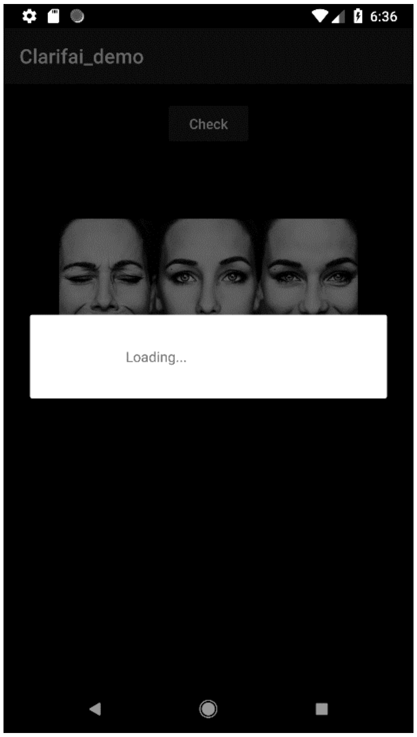
Project Management:

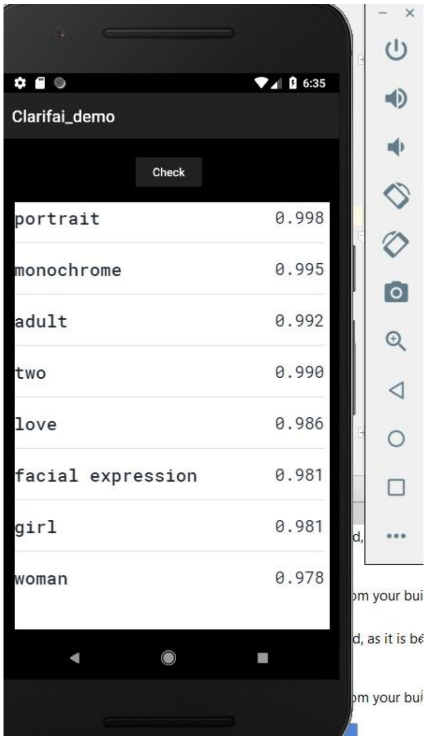
Work Completed:

- **Application with Clarifai API emulated on Android Phone:**



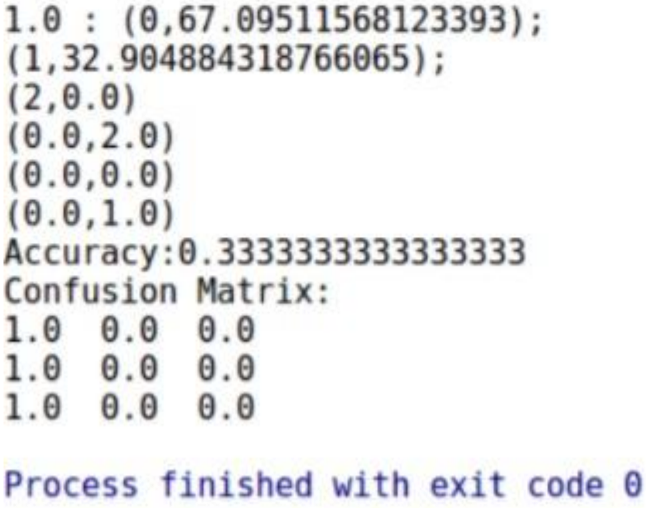






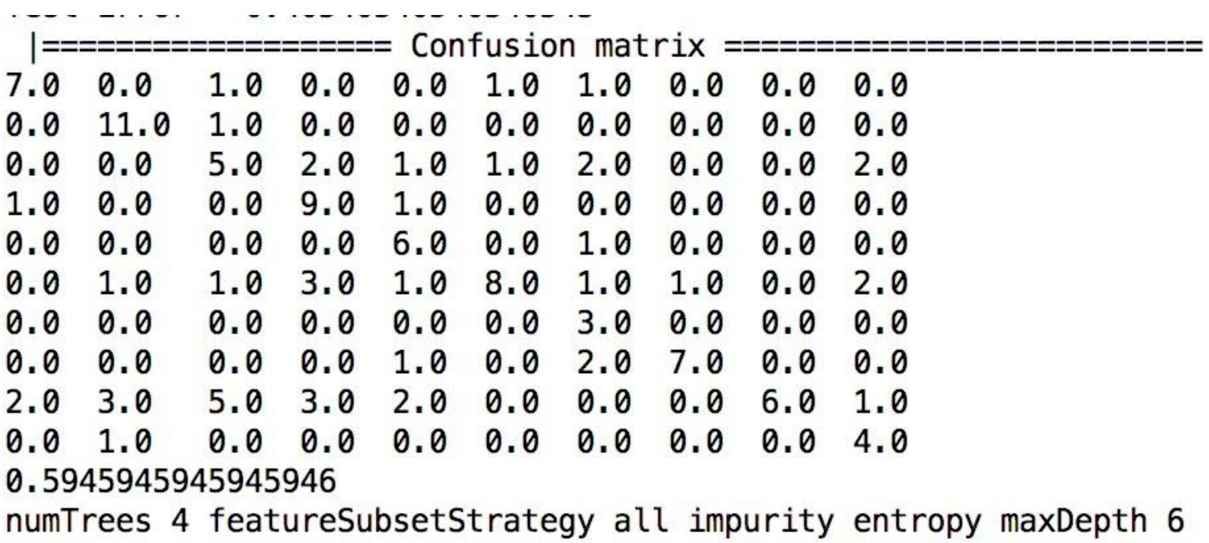
```
portrait               0.998
monochrome             0.995
adult                  0.992
two                    0.990
love                   0.986
facial expression      0.981
girl                   0.981
woman                  0.978
```

Accuracy achieved using Clarifai API:



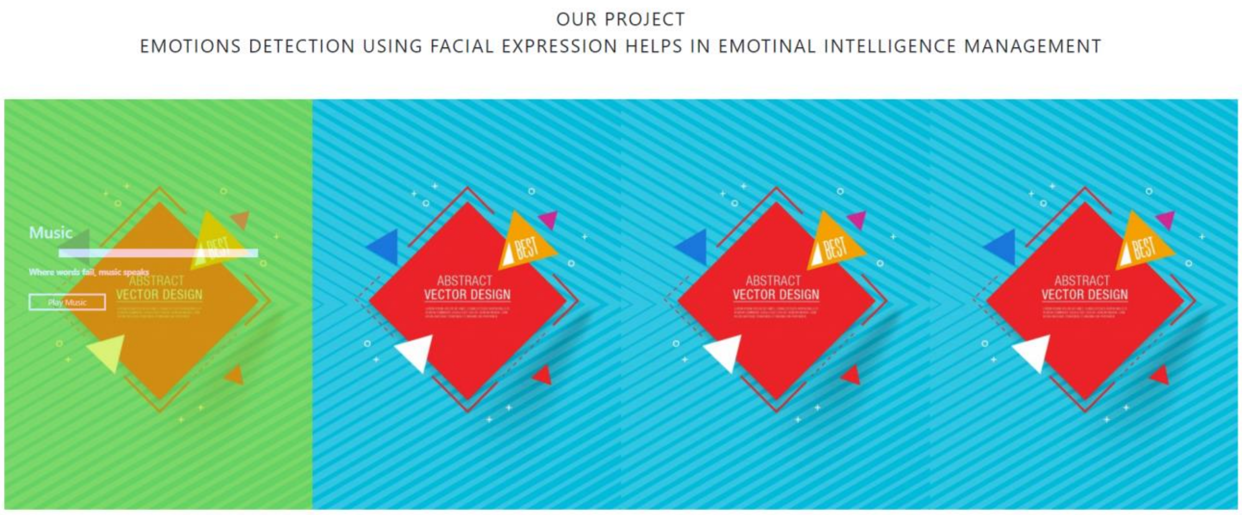
```
1.0 : (0,67.09511568123393);
(1,32.904884318766065);
(2,0.0)
(0.0,2.0)
(0.0,0.0)
(0.0,1.0)
Accuracy:0.3333333333333333
Confusion Matrix:
1.0  0.0  0.0
1.0  0.0  0.0
1.0  0.0  0.0

Process finished with exit code 0
```

Shallow Learning Approach:



```
 |================== Confusion matrix ==========================
7.0  0.0   1.0  0.0  0.0  1.0  1.0  0.0  0.0  0.0
0.0  11.0  1.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
0.0  0.0   5.0  2.0  1.0  1.0  2.0  0.0  0.0  2.0
1.0  0.0   0.0  9.0  1.0  0.0  0.0  0.0  0.0  0.0
0.0  0.0   0.0  0.0  6.0  0.0  1.0  0.0  0.0  0.0
0.0  1.0   1.0  3.0  1.0  8.0  1.0  1.0  0.0  2.0
0.0  0.0   0.0  0.0  0.0  0.0  3.0  0.0  0.0  0.0
0.0  0.0   0.0  0.0  1.0  0.0  2.0  7.0  0.0  0.0
2.0  3.0   5.0  3.0  2.0  0.0  0.0  0.0  6.0  1.0
0.0  1.0   0.0  0.0  0.0  0.0  0.0  0.0  0.0  4.0
0.5945945945945946
numTrees 4 featureSubsetStrategy all impurity entropy maxDepth 6
```

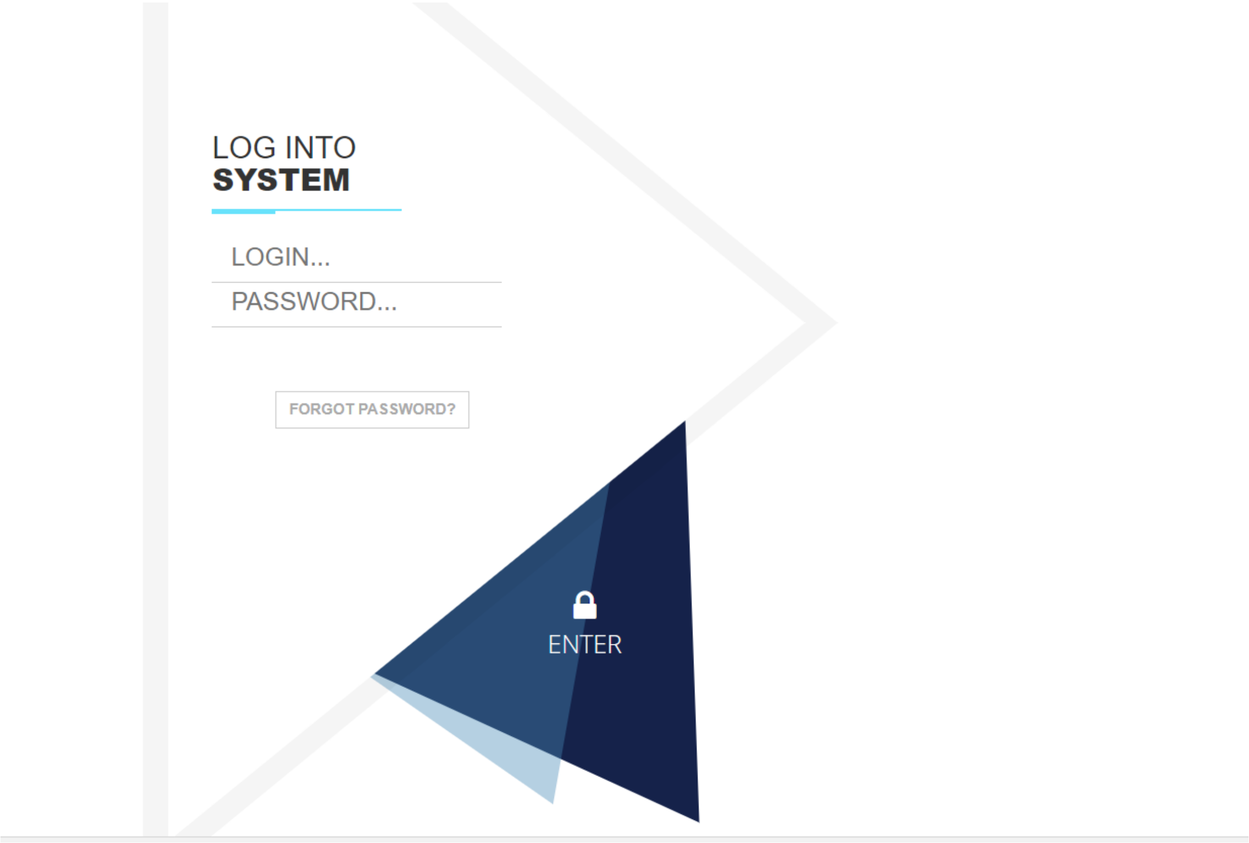
Web Application User Interface:



OUR PROJECT

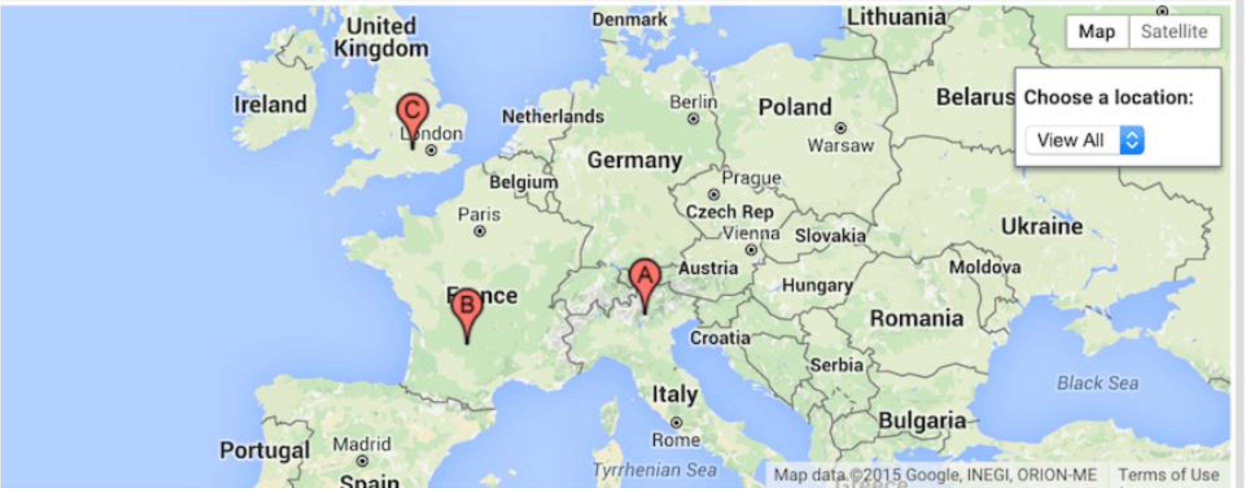
EMOTIONS DETECTION USING FACIAL EXPRESSION HELPS IN EMOTINAL INTELLIGENCE MANAGEMENT

Login UI:

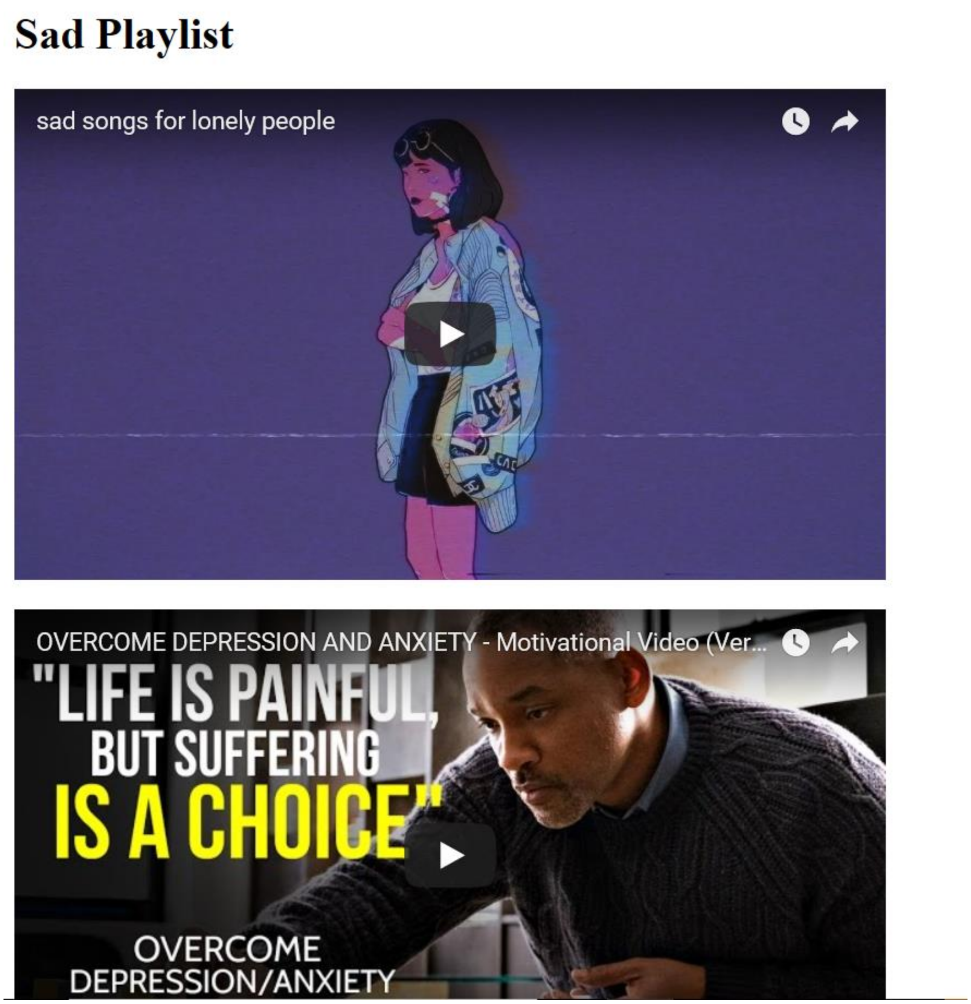


LOG INTO
SYSTEM

Maps API Embedded:



Videos Page UI:



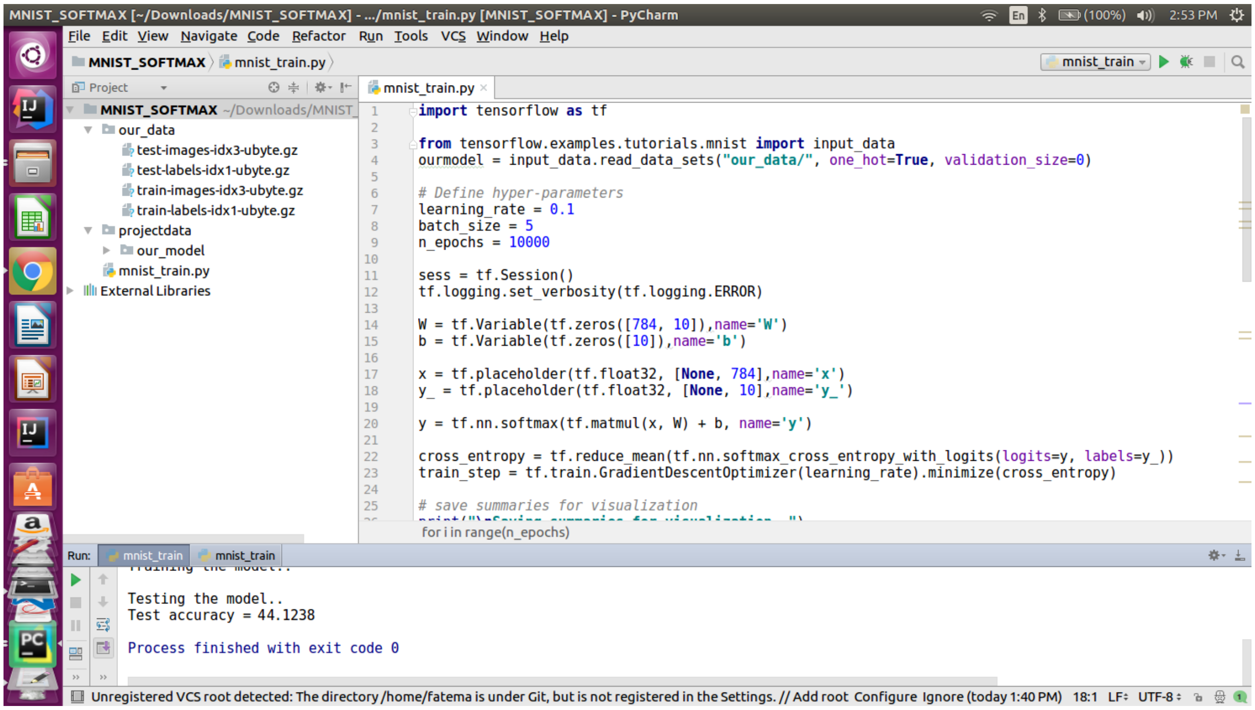
Sad Playlist

sad songs for lonely people

OVERCOME DEPRESSION AND ANXIETY - Motivational Video (Ver...

"LIFE IS PAINFUL, BUT SUFFERING IS A CHOICE"
OVERCOME DEPRESSION/ANXIETY

Accuracy achieved by Softmax Function :


MNIST_SOFTMAX [~/Downloads/MNIST_SOFTMAX] - .../mnist_train.py [MNIST_SOFTMAX] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
MNIST_SOFTMAX
mnist_train.py
Project
MNIST_SOFTMAX ~/Downloads/MNIST_
our_data
test-images-idx3-ubyte.gz
test-labels-idx1-ubyte.gz
train-images-idx3-ubyte.gz
train-labels-idx1-ubyte.gz
projectdata
our_model
mnist_train.py
External Libraries
Run: mnist_train mnist_train

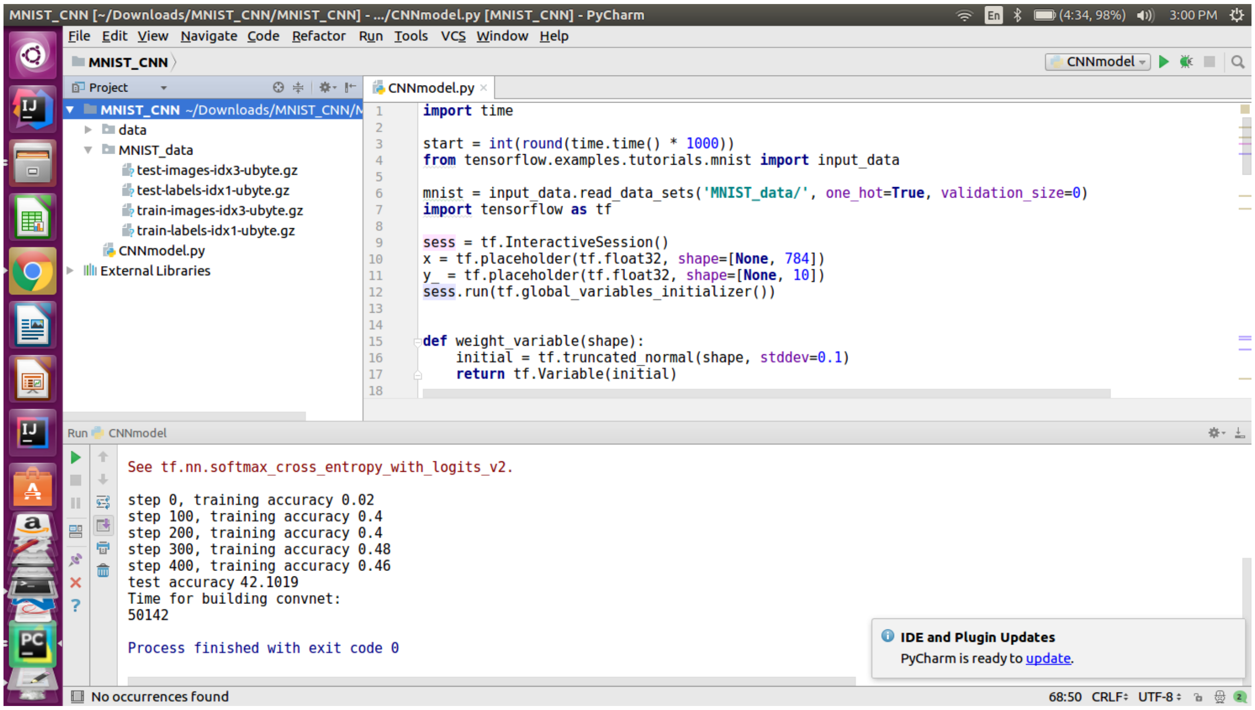
```
import tensorflow as tf

from tensorflow.examples.tutorials.mnist import input_data
ourmodel = input_data.read_data_sets("our_data/", one_hot=True, validation_size=0)

# Define hyper-parameters
learning_rate = 0.1
batch_size = 5
n_epochs = 10000

sess = tf.Session()
tf.logging.set_verbosity(tf.logging.ERROR)

W = tf.Variable(tf.zeros([784, 10]),name='W')
b = tf.Variable(tf.zeros([10]),name='b')

x = tf.placeholder(tf.float32, [None, 784],name='x')
y_ = tf.placeholder(tf.float32, [None, 10],name='y_')

y = tf.nn.softmax(tf.matmul(x, W) + b, name='y')

cross_entropy = tf.reduce_mean(tf.nn.softmax_cross_entropy_with_logits(logits=y, labels=y_))
train_step = tf.train.GradientDescentOptimizer(learning_rate).minimize(cross_entropy)

# save summaries for visualization
print("\nSaving summaries for visualization..")
```

for i in range(n_epochs)

```
Training the model..

Testing the model..
Test accuracy = 44.1238

Process finished with exit code 0
```

Unregistered VCS root detected: The directory /home/fatema is under Git, but is not registered in the Settings. // Add root Configure Ignore (today 1:40 PM) 18:1 LF: UTF-8:

Accuracy achieved by CNN Model :


MNIST_CNN [~/Downloads/MNIST_CNN/MNIST_CNN] - .../CNNmodel.py [MNIST_CNN] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
MNIST_CNN
CNNmodel
Project
MNIST_CNN ~/Downloads/MNIST_CNN/M
data
MNIST_data
test-images-idx3-ubyte.gz
test-labels-idx1-ubyte.gz
train-images-idx3-ubyte.gz
train-labels-idx1-ubyte.gz
CNNmodel.py
External Libraries
Run CNNmodel

```
import time

start = int(round(time.time() * 1000))
from tensorflow.examples.tutorials.mnist import input_data

mnist = input_data.read_data_sets('MNIST_data/', one_hot=True, validation_size=0)
import tensorflow as tf

sess = tf.InteractiveSession()
x = tf.placeholder(tf.float32, shape=[None, 784])
y_ = tf.placeholder(tf.float32, shape=[None, 10])
sess.run(tf.global_variables_initializer())


def weight_variable(shape):
    initial = tf.truncated_normal(shape, stddev=0.1)
    return tf.Variable(initial)
```

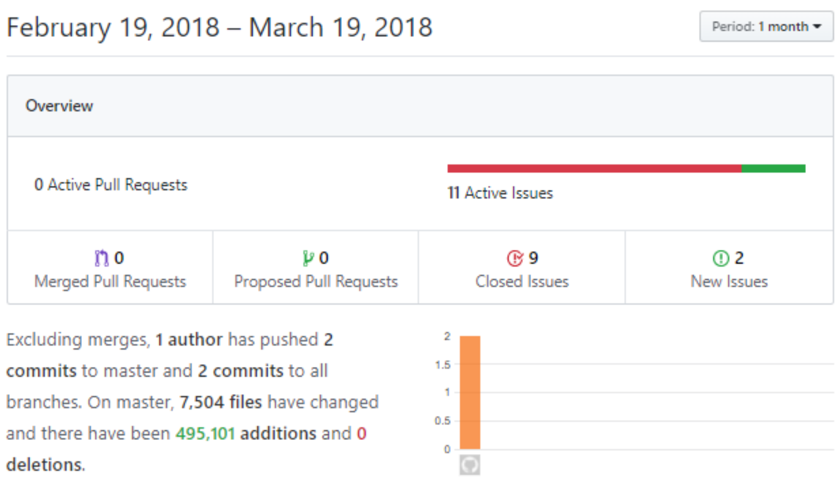
```
See tf.nn.softmax_cross_entropy_with_logits_v2.

step 0, training accuracy 0.02
step 100, training accuracy 0.4
step 200, training accuracy 0.4
step 300, training accuracy 0.48
step 400, training accuracy 0.46
test accuracy 42.1019
Time for building convnet:
50142

Process finished with exit code 0
```

IDE and Plugin Updates
PyCharm is ready to update.

No occurrences found 68:50 CRLF: UTF-8:

Zenhub Dashboard:

Overview of the progress:



February 19, 2018 – March 19, 2018

Overview

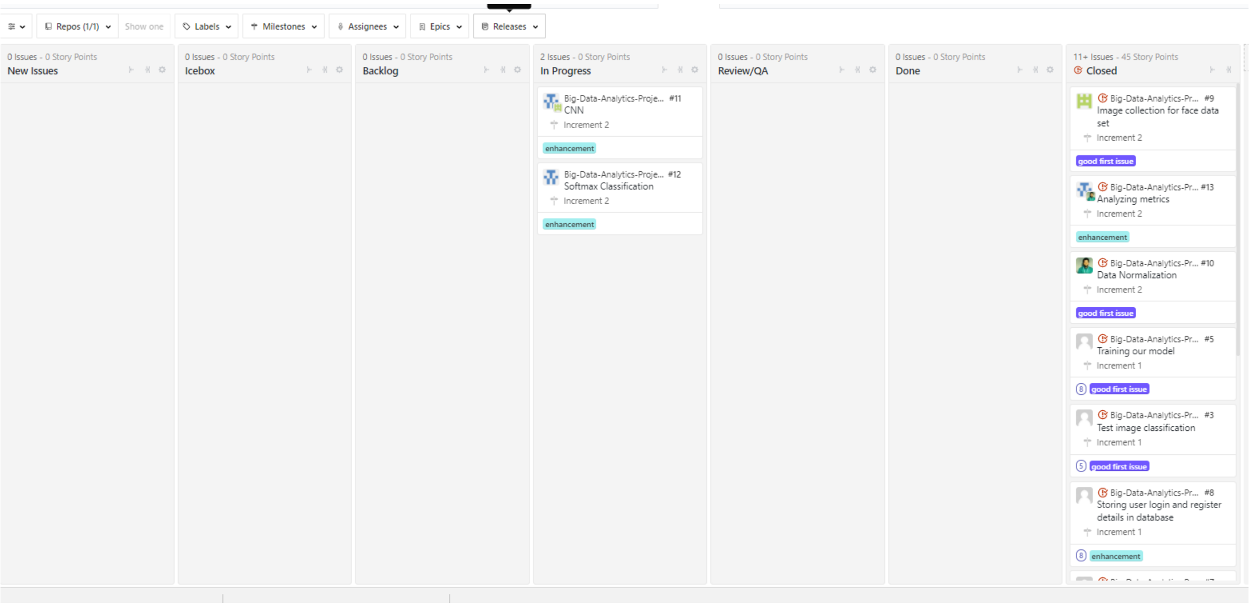
0 Active Pull Requests

11 Active Issues

0 Merged Pull Requests	0 Proposed Pull Requests	9 Closed Issues	2 New Issues

Excluding merges, **1 author** has pushed **2 commits** to master and **2 commits** to all branches. On master, **7,504 files** have changed and there have been **495,101 additions** and **0 deletions**.

Issue board:



0 Issues - 0 Story Points
New Issues

0 Issues - 0 Story Points
Icebox

0 Issues - 0 Story Points
Backlog

2 Issues - 0 Story Points
In Progress

Big-Data-Analytics-Proje... #11
CNN
Increment 2
enhancement

Big-Data-Analytics-Proje... #12
Softmax Classification
Increment 2
enhancement

0 Issues - 0 Story Points
Review/QA

0 Issues - 0 Story Points
Done

11+ Issues - 45 Story Points
Closed

Big-Data-Analytics-Pr... #9
Image collection for face data set
Increment 2
good first issue

Big-Data-Analytics-Pr... #13
Analyzing metrics
Increment 2
enhancement

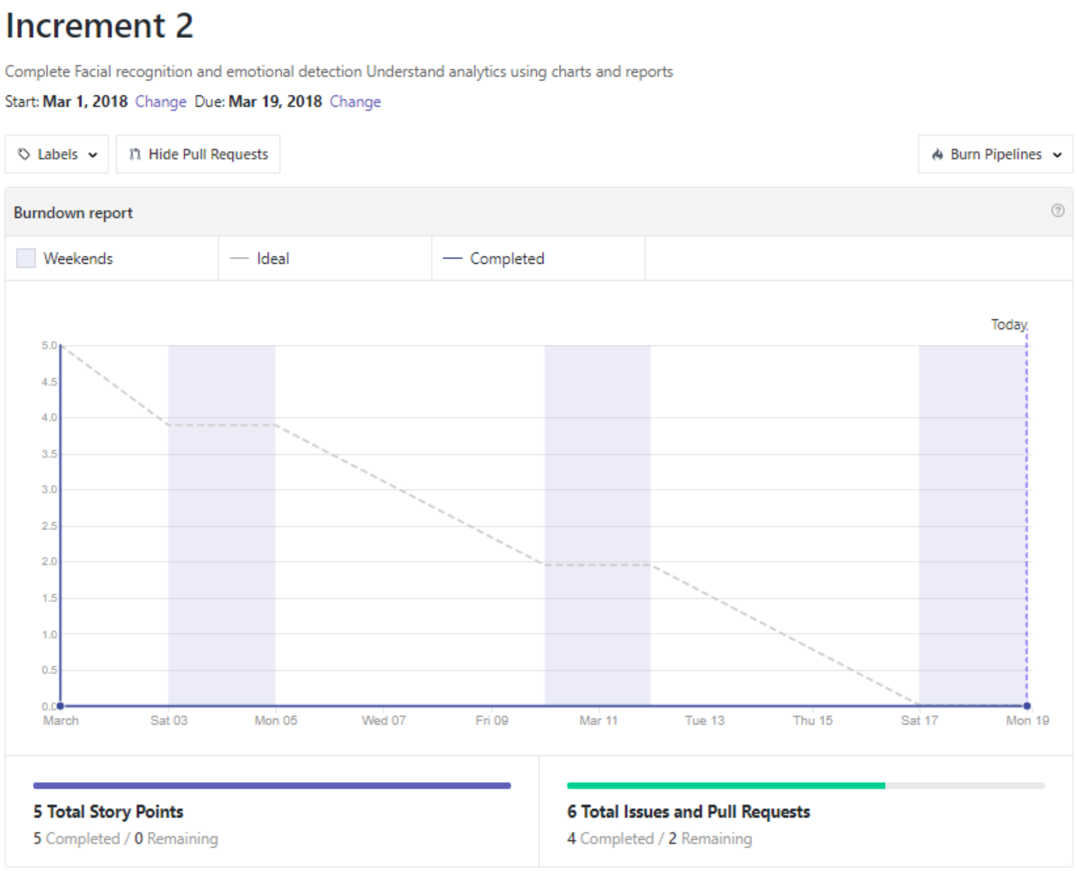
Big-Data-Analytics-Pr... #10
Data Normalization
Increment 2
good first issue

Big-Data-Analytics-Pr... #5
Training our model
Increment 1
8 good first issue

Big-Data-Analytics-Pr... #3
Test image classification
Increment 1
5 good first issue

Big-Data-Analytics-Pr... #8
Storing user login and register details in database
Increment 1
8 enhancement

Milestone Graph:



Increment 2

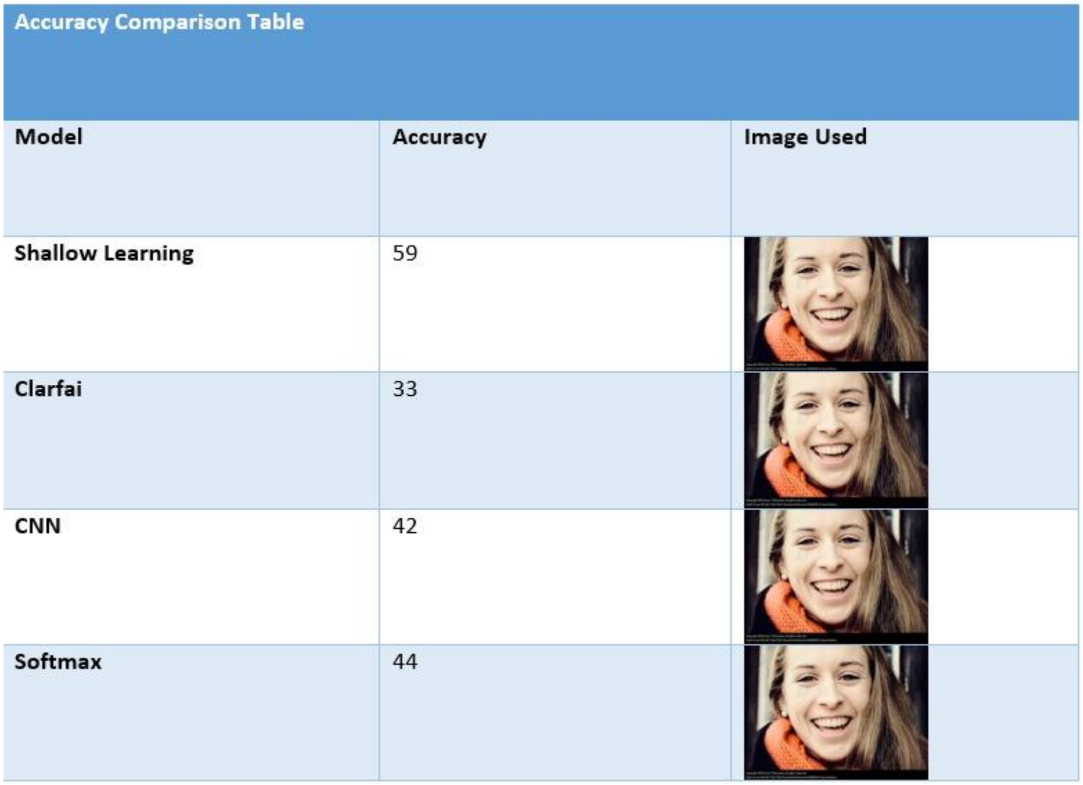
Complete Facial recognition and emotional detection Understand analytics using charts and reports

Start: **Mar 1, 2018** Change Due: **Mar 19, 2018** Change

5 Total Story Points
5 Completed / 0 Remaining

6 Total Issues and Pull Requests
4 Completed / 2 Remaining

Accuracy Table showing the accuracies achieved using different techniques for Increment II:

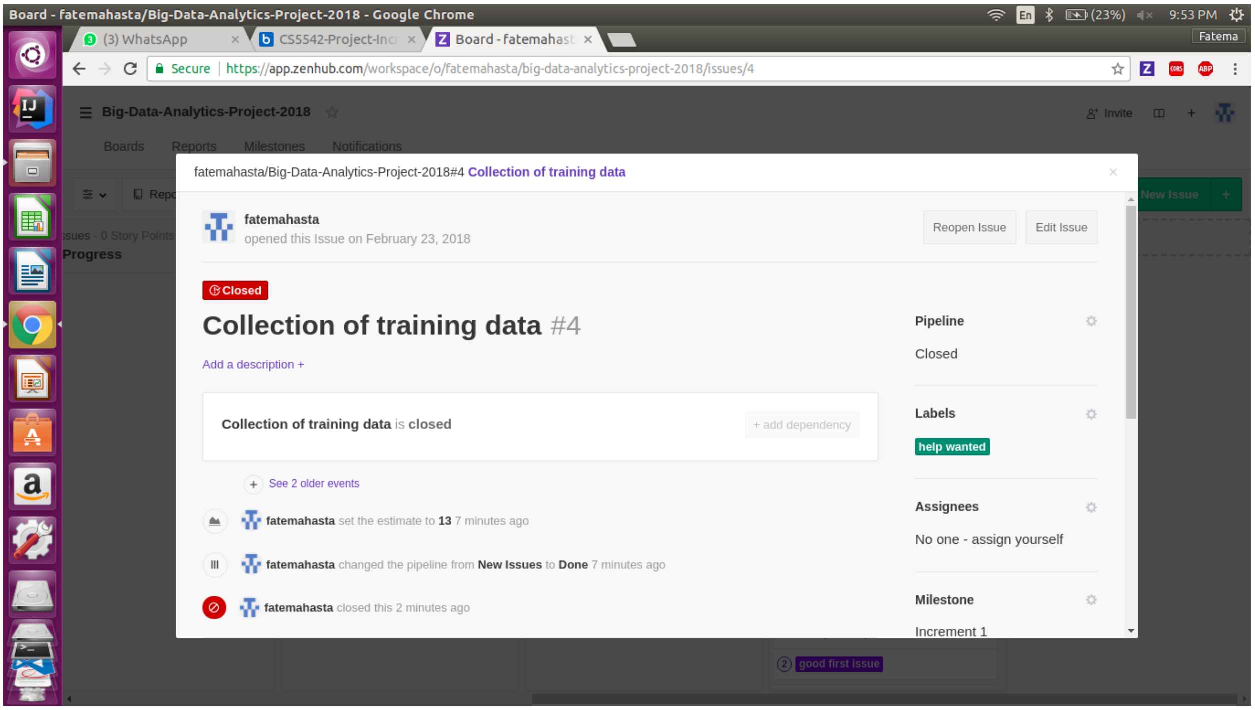


Accuracy Comparison Table		
Model	**Accuracy**	**Image Used**
Shallow Learning	59	
Clarfai	33	
CNN	42	
Softmax	44	

Below are the screenshots of the work done by Project Increment I:

- **Collected Training data**

We took some images of people’s faces and videos of people expressing emotions.


Board - fatemahasta/Big-Data-Analytics-Project-2018 - Google Chrome
(3) WhatsApp
CS5542-Project-Incr
Board - fatemahast
Secure
https://app.zenhub.com/workspace/o/fatemahasta/big-data-analytics-project-2018/issues/4
Big-Data-Analytics-Project-2018
Boards
Reports
Milestones
Notifications
Invite
New Issue
Reopen Issue
Edit Issue
+ add dependency

fatemahasta/Big-Data-Analytics-Project-2018#4 **Collection of training data**

fatemahasta
opened this Issue on February 23, 2018

Closed

Collection of training data #4

Add a description +

Collection of training data is closed

See 2 older events

fatemahasta set the estimate to **13** 7 minutes ago

fatemahasta changed the pipeline from **New Issues** to **Done** 7 minutes ago

fatemahasta closed this 2 minutes ago

Pipeline
Closed

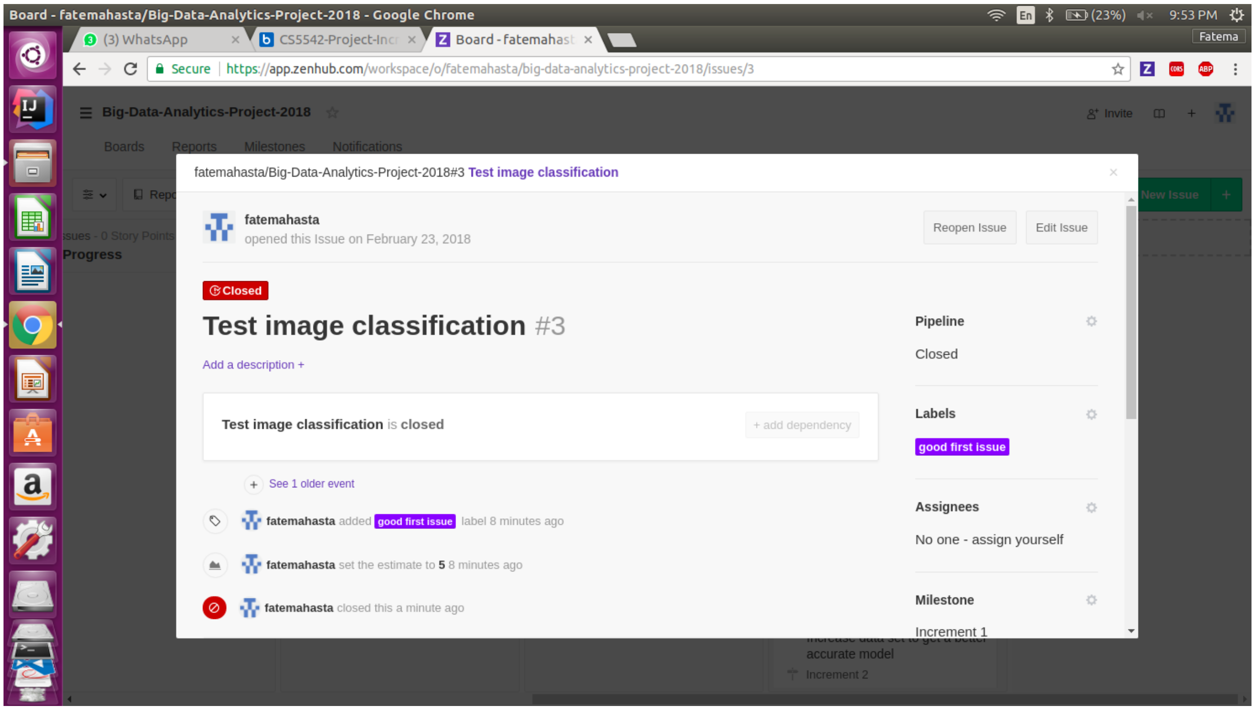
Labels
help wanted

Assignees
No one - assign yourself

Milestone
Increment 1

- **Image Classification**

  Classified these images into classes of emotions like ‘Happy’, ‘Sad’, ‘Angry’, ‘Surprised’, ‘Bored’.



fatemahasta/Big-Data-Analytics-Project-2018#3 Test image classification

fatemahasta
opened this Issue on February 23, 2018

Closed

Test image classification #3

Test image classification is closed

fatemahasta added good first issue label 8 minutes ago

fatemahasta set the estimate to 5 8 minutes ago

fatemahasta closed this a minute ago

Pipeline
Closed

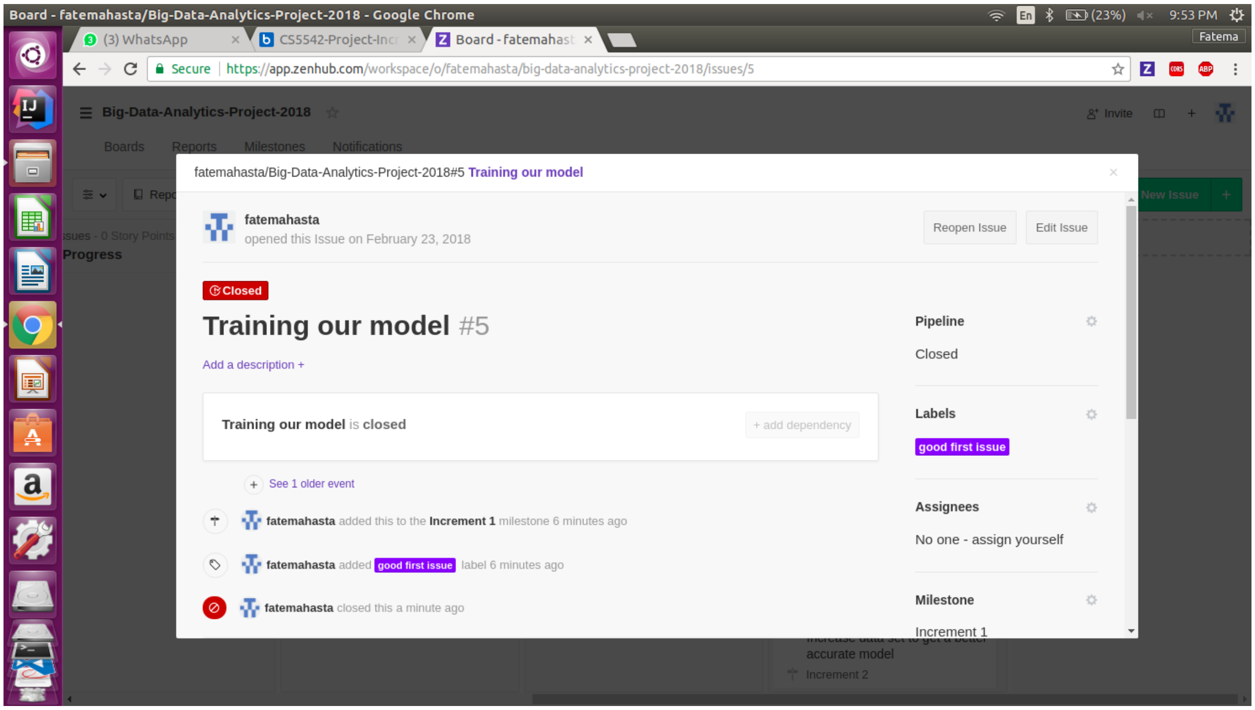
Labels
good first issue

Assignees
No one - assign yourself

Milestone
Increment 1

- **Trained the Model**

  We put some images in the training data and classified the classes. If the input test image matches with the training data; the emotion class will be omitted



Board - fatemahasta/Big-Data-Analytics-Project-2018 - Google Chrome

https://app.zenhub.com/workspace/o/fatemahasta/big-data-analytics-project-2018/issues/5

fatemahasta/Big-Data-Analytics-Project-2018#5 Training our model

fatemahasta
opened this Issue on February 23, 2018

Closed

Training our model #5

Add a description +

Training our model is closed

+ See 1 older event

fatemahasta added this to the Increment 1 milestone 6 minutes ago

fatemahasta added good first issue label 6 minutes ago

fatemahasta closed this a minute ago

Pipeline: Closed

Labels: good first issue

Assignees: No one - assign yourself

Milestone: Increment 1

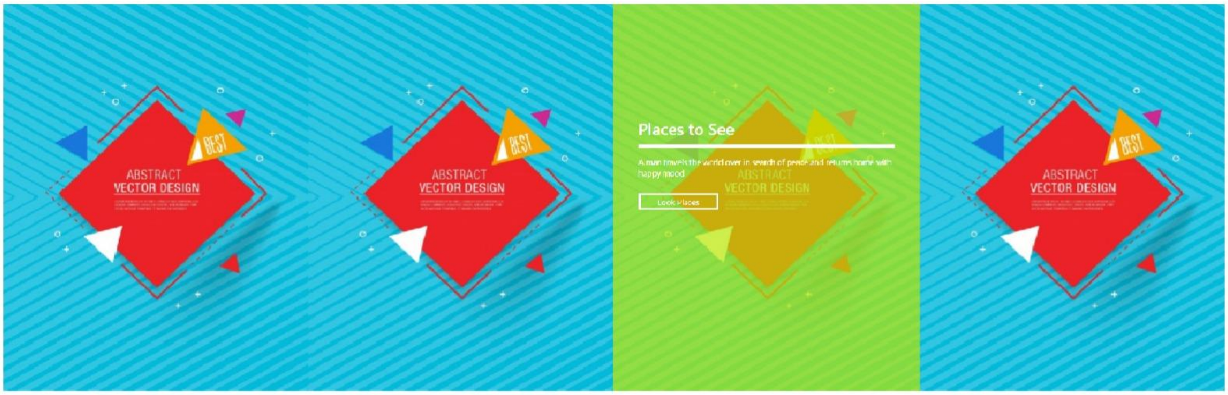
- **Static User Interface**

  We made a webpage: which will make the users to interact with the application.

  One page lets us to select one image from the collection of images displayed.

  Next page is a dashboard which is unique for different class of emotions. And will have buttons to access different options like Music, Videos, Places & Tips to manage the emotion.

Image of the Dashboard:



Places to See

A man travels the world over in search of peace and returns home with happy mood

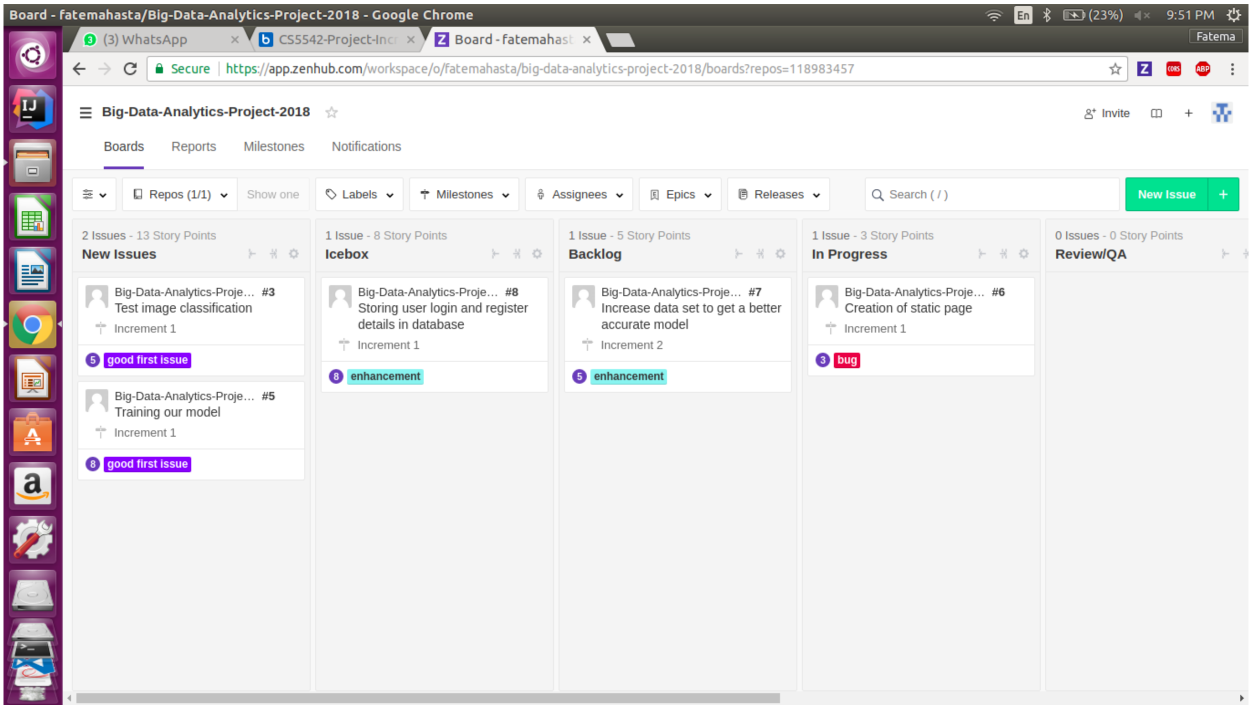
- Acquired Confusion Matrix

  We used different calculation techniques: Random Forest, Decision Tree and acquired the confusion matrix.

Time Taken: Each of us worked for around 10 hours each.

Contributors: All of us worked towards this aim equally. No particular task is achieved by only one. But, we split the work equally overall.

Dashboard:



Board - fatemahasta/Big-Data-Analytics-Project-2018 - Google Chrome

https://app.zenhub.com/workspace/o/fatemahasta/big-data-analytics-project-2018/boards?repos=118983457

Big-Data-Analytics-Project-2018

2 Issues - 13 Story Points
New Issues

Big-Data-Analytics-Proje… **#3**
Test image classification
Increment 1
5 good first issue

Big-Data-Analytics-Proje… **#5**
Training our model
Increment 1
8 good first issue

1 Issue - 8 Story Points
Icebox

Big-Data-Analytics-Proje… **#8**
Storing user login and register details in database
Increment 1
8 enhancement

1 Issue - 5 Story Points
Backlog

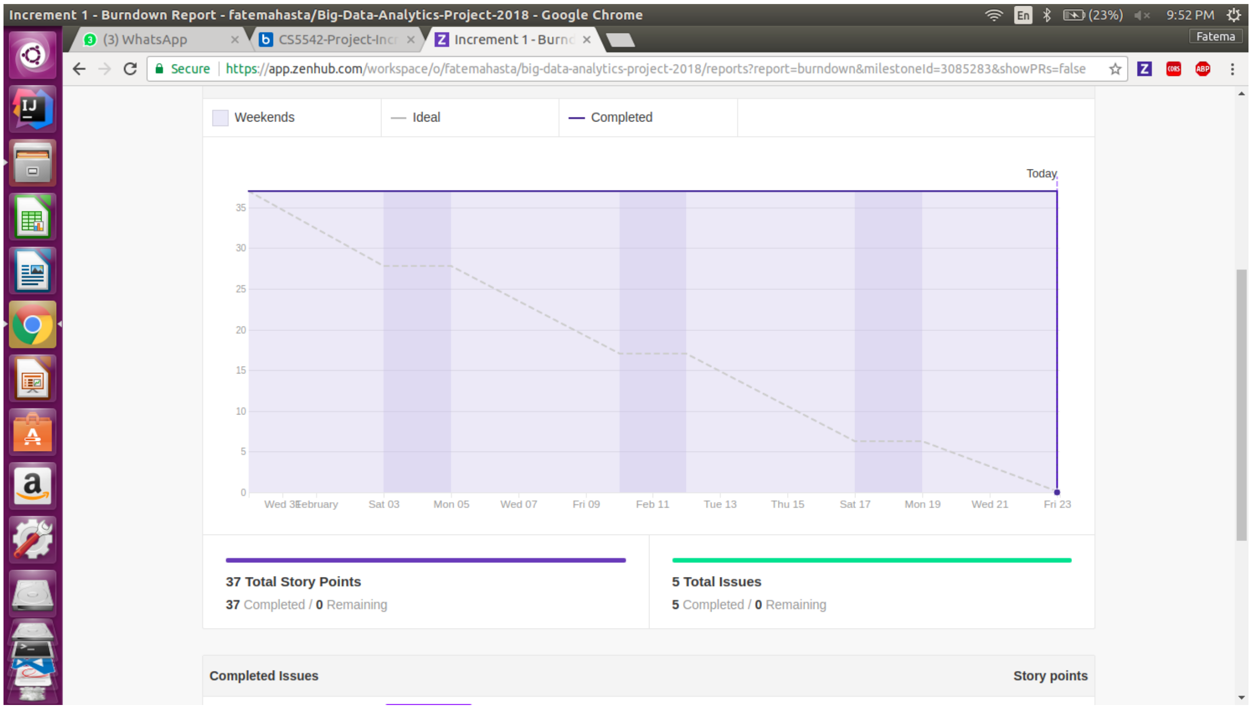
Big-Data-Analytics-Proje… **#7**
Increase data set to get a better accurate model
Increment 2
5 enhancement

1 Issue - 3 Story Points
In Progress

Big-Data-Analytics-Proje… **#6**
Creation of static page
Increment 1
3 bug

0 Issues - 0 Story Points
Review/QA

Burndown Chart:



Increment 1 - Burndown Report - fatemahasta/Big-Data-Analytics-Project-2018 - Google Chrome

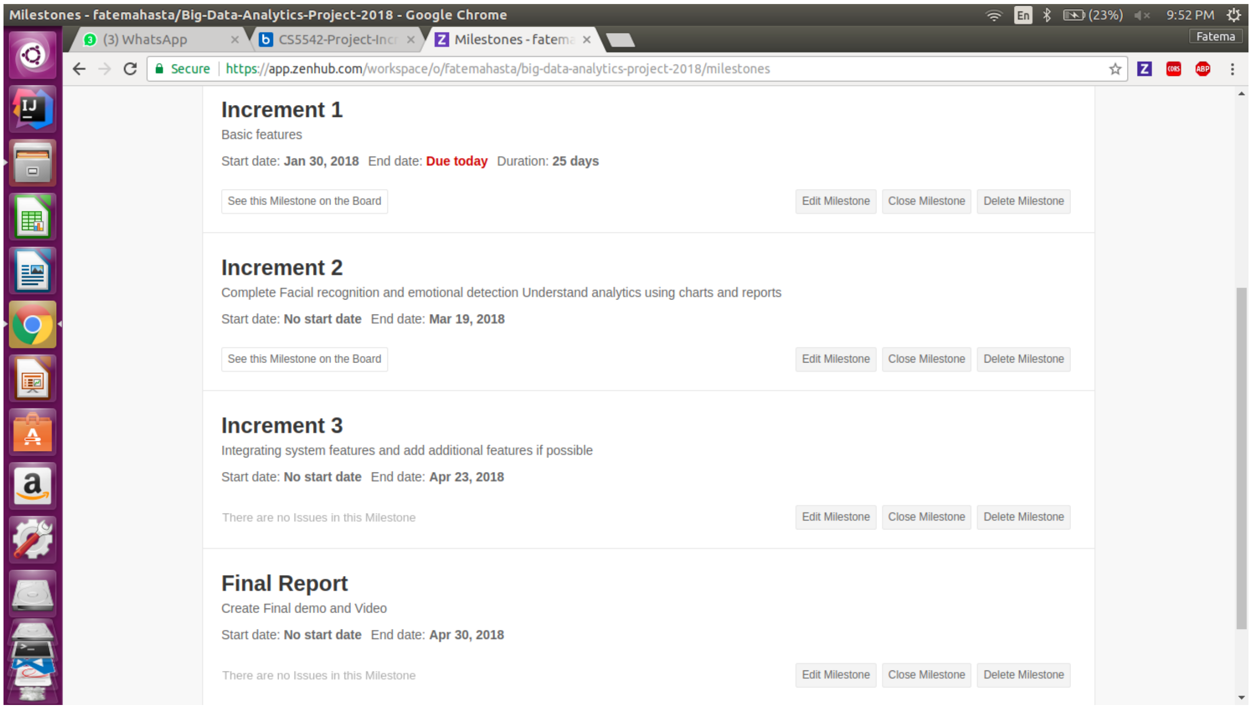
https://app.zenhub.com/workspace/o/fatemahasta/big-data-analytics-project-2018/reports?report=burndown&milestoneId=3085283&showPRs=false

37 Total Story Points
37 Completed / **0** Remaining

5 Total Issues
5 Completed / **0** Remaining

Completed Issues **Story points**

Work to be completed: Increment 3


Milestones - fatemahasta/Big-Data-Analytics-Project-2018 - Google Chrome
(3) WhatsApp
CS5542-Project-Incr
Milestones - fatema
9:52 PM
(23%)
Fatema
Secure
https://app.zenhub.com/workspace/o/fatemahasta/big-data-analytics-project-2018/milestones

Increment 1

Basic features

Start date: **Jan 30, 2018** End date: **Due today** Duration: **25 days**

See this Milestone on the Board | Edit Milestone | Close Milestone | Delete Milestone

Increment 2

Complete Facial recognition and emotional detection Understand analytics using charts and reports

Start date: **No start date** End date: **Mar 19, 2018**

See this Milestone on the Board | Edit Milestone | Close Milestone | Delete Milestone

Increment 3

Integrating system features and add additional features if possible

Start date: **No start date** End date: **Apr 23, 2018**

There are no Issues in this Milestone | Edit Milestone | Close Milestone | Delete Milestone

Final Report

Create Final demo and Video

Start date: **No start date** End date: **Apr 30, 2018**

There are no Issues in this Milestone | Edit Milestone | Close Milestone | Delete Milestone

- Integrating the web application with the BDA Model.
- Implementing Google Charts API & any of the commercial Music APIs.
- TensorFlow Implementation.

Time to be taken: 40 Hours

Issues/Concerns:

- As we are dealing with facial expressions, it requires a big deal of images for training and still after a lot of training there is always some room for improvement. Achieving high accuracy is not possible all the times, it depends on the image of the test data.

Link to the source code: https://github.com/fatemahasta/Big-Data-Analytics-Project-2018/tree/master/Project%20Increment%202