

# **Monument Recognition**



By,

Team -5

Panja,Kumara Satya Gopal(kpqmd@mail.umkc.edu)

Linga,Sivarama Krishna(slhx4@mail.umkc.edu)

Vikesh Padarthi (vpn7d@mail.umkc.edu)

Vundela,Karthik Reddy(kvxc5@mail.umkc.edu)

**Project Report -1**

**Big Data Analytics and Applications**

## **1 Introduction:**

Nowadays finding the matching of images in order to establish a measure of similarity is a major problem. Here we introduce a “Monument Recognition System”. It takes monument image as input from user and gives description about that images as output. For this first we want create our own data set for different monuments. Based on different algorithms, we can find input belongs to which category. Based on that monument category and apis we can give description about that monument.

## **2 Project Objectives:**

### **2.1 Significance:**

Whenever we visit a place we not only want to take the pictures in front of it but we would like to know history about that place. So instead of relying on some guide or go through the Google and search, a simple website which will help to know about the place in details in both text and audio format. Our objective is to develop a simple website where for the website we will accept input as a image which might be a statue or a building. we will learn about the image and know what it is like statue of liberty and we will get the details of that image like who developed and where it is etc and we will display about it. And we want to turn this text into voice and user should be able to listen to it.

### **2.2 Features:**

- User can open the application on web browser and see the home page. On clicking on home page user will see signin and signup pages.
- User needs to create an account by clicking on signup page.
- User can login by using his credentials
- On Successful login we navigate user to main page.
- In main page, user can upload an image.
- User can learn about the monument
- User can find related places
- User can hear the speech about monument.

## **3 Approach**

### **3.1 Data Sources:**

Here we can create our own data set for different monuments. And we can use google and some apis for retrieving data about monuments.

### **3.2 Analytical Tools:**

Clarifai API:

This is used to detect what is present in a given. We use the monument dataset and the Clarifai API for detecting what is present in the image.

Spark(ML Library):

The Spark machine learning library is used for training the the image data set with various machine learning algorithms such as k-means etc which are in-built in the Spark ML library and this can later be used to test the images after we get a trained model.

Tensor Flow:

Tensor Flow is also being used for the image classification purpose.

### 3.3 Analytical Tasks

Initially the images are used for building the model. This is called the training phase. After building the model testing can be done to get the information about image. The input image is first analysed for extracting the key features. Then the histograms are built from these key features which represent the bag of words present in the image. This is then used as input for training the model. After the model is finally is tested with various sets of inputs and the process is repeated by tuning the parameters until desired accuracy and acceptable error rate, precision, recall and f-measure are obtained. After the model gets ready it can be used for the end-user app. In addition this the Clarifai API and tensor flow are also used to get better results.

### 3.4 Expected Inputs/Outputs:

When we finish developing this system, it should be able to provide movies recommendation to users. Based on user preferences, out system will recommend movies to users.

Monument Recognition

User Name

Password

Login

When the user opens our application, first it shows login page. User need to create an account. By using the user credentials, he can login into the application. On successful login, it will navigate into main page.



### **3.5 Algorithms**

There are several algorithms that can be used for the purpose of analysing the data sets.

However, based on the application appropriate algorithms need to be chosen. For example, Naive Bayes performs better when the data set is a text data. On the other hand, image data gets better results when k-means is used. Hence we used K-means for classifying the image.

Initially, the number of clusters and iterations are predefined, and the given dataset is clustered into different clusters based on the common attributes. Then the model is built based on the training data. After that, testing can also be done.

We also used random forest algorithm for image classification. Random forest proves to be more advantageous when compared to decision tree as decision tree involves only one tree whereas random forest is an ensemble of decision trees in parallel thereby giving better accuracy and lesser error rate.

## **4 Related Work:**

### **4.1 Open Source Projects:**

The image classification model in the project has been inspired from the model below. It has several programs for classification of images.

<https://github.com/ntetor/Image-Classifier>

A list of projects along with guidelines to do them has been provide in the below link which were all used as a part of final year projects

[http://www.di.ens.fr/willow/teaching/recvis09/final\\_project/](http://www.di.ens.fr/willow/teaching/recvis09/final_project/)

### **4.2 Literature Reviews:**

Two pairs also helped in getting an idea of what image classification is about. The first gives a clean account of the process that has to be adopted for doing any image classification.

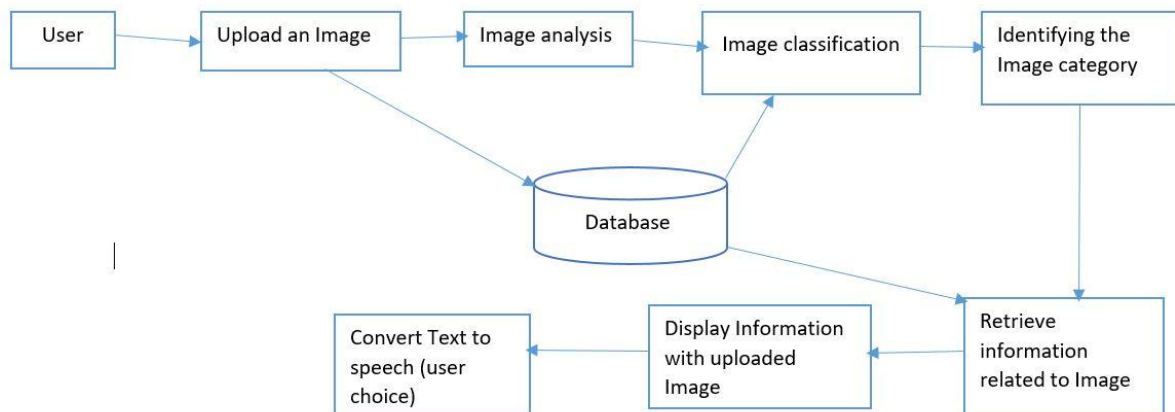
The second one is more subjective I.e. it provides a way of distinguishing images (which is explicit from the name of the paper!). <http://cs229.stanford.edu/proj2013/KrzesinskiWilder-ImageObjectClassification.pdf>

[https://sites.ualberta.ca/~bang3/files/DogCat\\_report.pdf](https://sites.ualberta.ca/~bang3/files/DogCat_report.pdf)

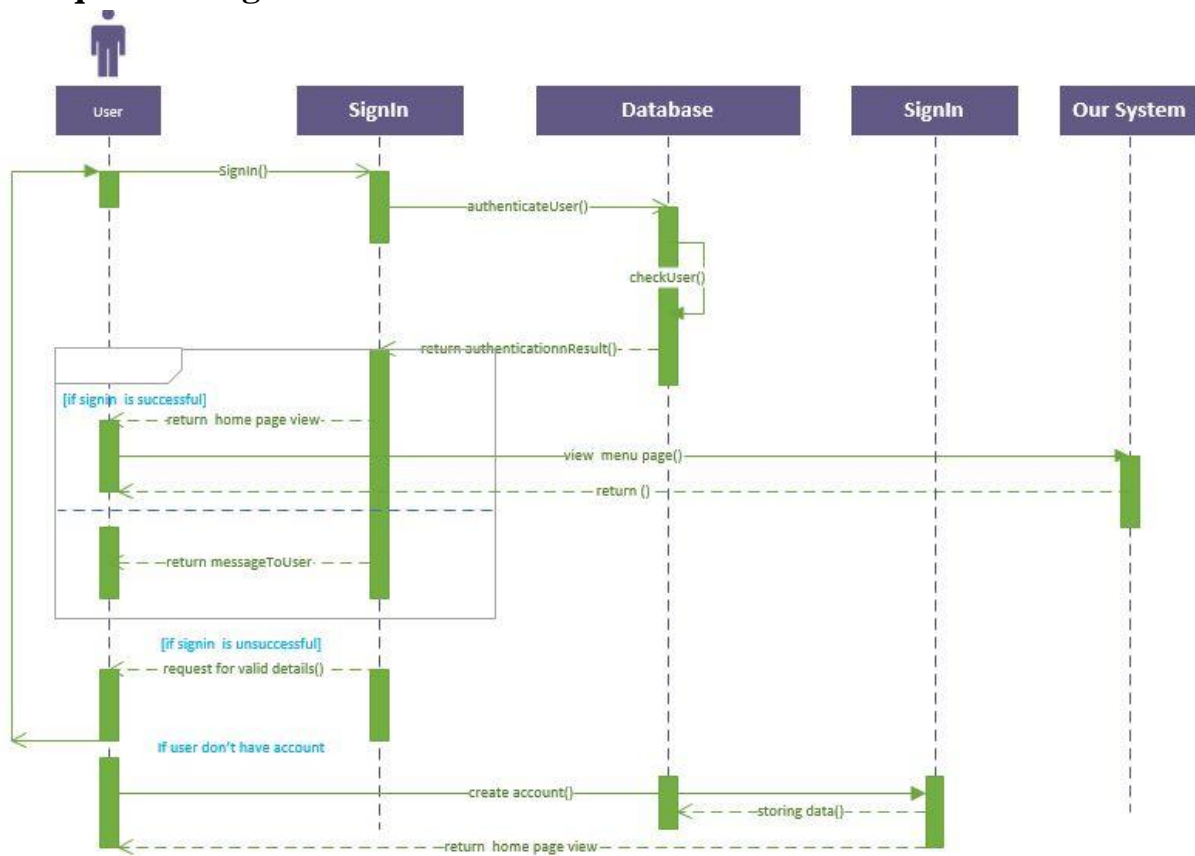
## **5 Application Specification:**

### **5.1 Software Architecture:**

### **5.2 Work flow Diagram:**



### 5.3 Sequence Diagram:



## 6. Implementation:

### 6.1 Clarifai API outputs:

```

21
22 public class ImageAnnotation {
23     public static void main(String[] args) throws IOException {
24         final ClarifaiClient client = new ClarifaiBuilder( appId: "KKQIegBW9u0l_3vaMSzqq4QCfPNyNBvB7XNBz:

```

Run: ClarifaiExample ImageAnnotation

```

/usr/lib/java/jdk1.8.0_121/bin/java ...
mammal - 0.9961703
animal - 0.9665432
no person - 0.9605458
wildlife - 0.9501431
elephant - 0.91439927
group - 0.9048673
two - 0.90253234
nature - 0.8836016
cute - 0.86116207
people - 0.8519159
funny - 0.84840024
one - 0.8444712
bull - 0.83605736
zoo - 0.83487666
wild - 0.8283373
safari - 0.80364776
grass - 0.80207586
illustration - 0.8003653
outdoors - 0.79099995
domestic - 0.78200275

Process finished with exit code 0

```

```

24         final ClarifaiClient client = new ClarifaiBuilder( appId: "KKQIegBW9u0l_3vaMSzqq4QCfPNyNBvB7XNBz:

```

Run: ClarifaiExample ImageAnnotation

```

/usr/lib/java/jdk1.8.0_121/bin/java ...
*****output/mainframes/0_0.8747855917667239.jpg*****
architecture - 0.9961723
travel - 0.9811857
building - 0.9724196
religion - 0.9615941
tourism - 0.9469688
no person - 0.9427623
ancient - 0.94246465
minaret - 0.9360717
city - 0.93596053
sky - 0.93429244
old - 0.93328273
traditional - 0.93052924
arch - 0.92968416
culture - 0.928947
dome - 0.9283396
marble - 0.9111028
monument - 0.9019058
outdoors - 0.89785147
castle - 0.89574546
religious - 0.8882282

```





[Secure](#) | <https://developers.google.com/actions/web-simulator>

Actions on Google    Design    **Develop**    Distribute    Support

GUIDES    SAMPLES    REFERENCE    **WEB SIMULATOR**

What's new in the Actions on Google Web Simulator? The Actions on Google Home, starting with the latest release, is designed to help you test your actions as closely as possible to how users will experience your actions and can catch many issues and bugs that delay approval. Login to your Google Home with the same account that you are deploying your actions to test on the device.

### Dialog

- what is your project?
- We are team 5. And our project is monument recognition
- tajmahal
- The Taj Mahal is an ivory-white marble mausoleum on the south bank of the Yamuna river in the Indian city of Agra. It was commissioned in 1632 by the Mughal emperor, Shah Jahan to house the tomb of his favourite wife, Mumtaz Mahal. The tomb is the centerpiece of a 17-hectare complex, which includes a mosque and a guest house, and is set in formal gardens bounded on three sides by a crenellated wall.
- statue of liberty
- The Statue of Liberty is a colossal neoclassical sculpture on Liberty Island in New York Harbor in New York City, in the United States. The copper statue, a gift

Type something or click the mic and speak

### Log

Action Package    02/27/2017 @ 5:59PM

Request

```
{
  "query": "statue of liberty",
  "accessToken": "ya29.Glt_A4cFADUjQvssugL3DmBqPFRZ24yVdSDBL7kNmr1e145771PSA-PVYlorgY3s1KvDL_0TJ3K1YpDTGAGDFX77rLUxyGyISPAVGQWETskkew",
  "expectUserResponse": true,
  "conversationToken": "C12DI2U4Y...",
  "debugInfo": {
    "assistantToAgentDebug": {
      "assistantToAgentJson": {
        "user": {
          "userId": "vivvnetPV3CANTRFSQTs1ehHwQBFerFA8FXV4BCJLE="
        },
        "conversation": {
          "conversationId": "1488239945916",
          "type": 2,
          "conversationToken": "[]"
        },
        "inputs": [
          {
            "intent": "Assistant.intent.action.TEXT",
            "rawInputs": [
              {
                "input_type": 2,
                "query": "statue of liberty",
                "annotation_sets": []
              }
            ],
            "arguments": [
              {
                "name": "text",
                "raw_text": "statue of liberty",
                "text_value": "statue of liberty"
              }
            ]
          }
        ],
        "surface": {
          "capabilities": [
            {
              "name": "actions.capability.AUDIO_INPUT"
            }
          ]
        }
      }
    }
  }
}
```





---









8



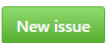




| Class Id | Name                     | Responsibility  |
|----------|--------------------------|---|
| 32       | Panja, Kumara Satya Goal | Google Conversation API, Designing UML diagrams and Expected outcomes |
| 21       | Linga, Siva Rama Krishna | Clarifai API, Workflow Diagram, Data collection                       |
| 30       | Padarathi, Vikesh        | Architecture Diagram, User Interface, Testing                         |
| 43       | Vundela, Karhik          | Project Management, Gathering Requirements, Documentation             |





































## 7.2 Issues:

KarthikVundela / BigDataAnalyticsandAppsCS5542\_Project   Unwatch 3  Star 0  Fork 0

 Code  Issues 9  Pull requests 0  Boards  Reports  Projects 0  Wiki 

Filters   Labels  Milestones  New issue

 9 Open  0 Closed Author Labels Milestones Assignee Sort

-   Clarifai API  
#9 opened just now by KarthikVundela  Increment 1 
-   Designing User Interface  
#8 opened a minute ago by KarthikVundela  Increment 1 
-   Documentation  
#7 opened 2 minutes ago by KarthikVundela  Increment 1 
-   Google Conversation API  
#6 opened 3 minutes ago by KarthikVundela  Increment 1 
-   System Architecture and Related Work  
#5 opened 3 minutes ago by KarthikVundela  Increment 1 
-   Project Management  
#4 opened 4 minutes ago by KarthikVundela  Increment 1 
-   Workflow diagram  
#3 opened 4 minutes ago by KarthikVundela  Increment 1 
-   Designing UML diagrams  
#2 opened 5 minutes ago by KarthikVundela  Increment 1 
-   Gathering Requirements  
#1 opened 7 minutes ago by KarthikVundela  Increment 1 

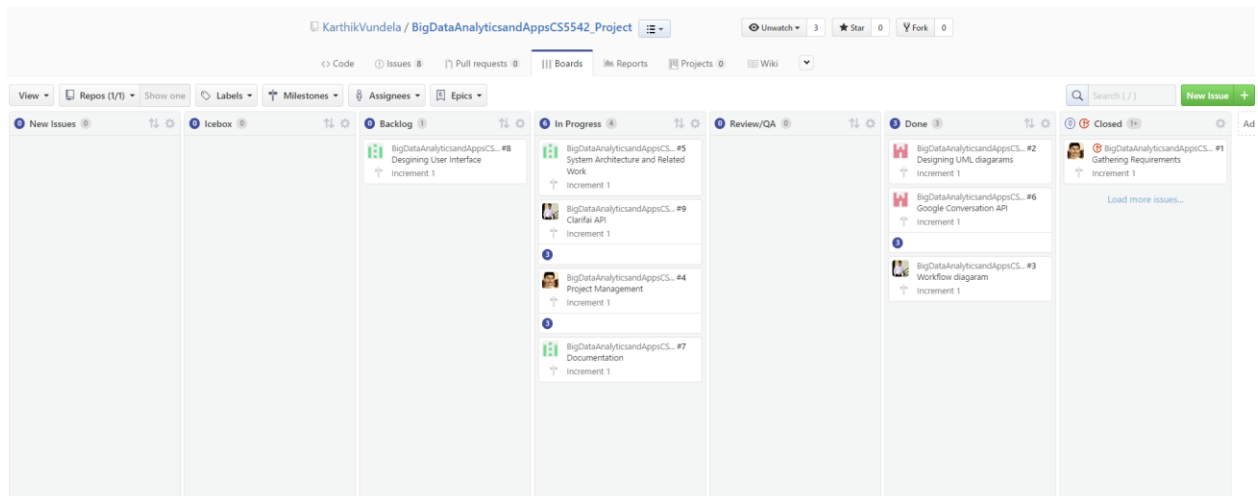
## 7.3 Burndown Chart of Increment 1:



Remaining Issues and Pull Requests

Story points

## Zenhub Board:



## Work Completed:

Google Conversation, Clarifai API, Designing UML Diagrams, Software Architecture, Gathering Requirements

**Work need to be Completed:**

Dataset collection, Google Cardboard API, Machine Learning Application

**8 References:**

Tutorial code 3,5

<https://github.com/ntetor/Image-Classifier>

[http://www.di.ens.fr/willow/teaching/recvis09/final\\_project/](http://www.di.ens.fr/willow/teaching/recvis09/final_project/)