VQA- Visual Question Answering

#5 Jada, Santosh | #11 Mehta, Avni | #14 Muppala, Raji | #19 Patel, Harshil

1.1 Abstract

There are several projects on visual question answering; We are taking MS COCO data which got 80,000 images with different domain. We ae limiting our VQA model to interiors (bathroom,bedroom,kitchen, living). Build models using Machine learning and Deep Learning algorithms.

This application will include a web application that enables the user to either select an image from the available images or upload an image. The user can then ask a question about the selected image. The main purpose of this application is to provide an accurate natural language answer in real-time. This is based on show and tell model.

1.2 Approach

Segregate Interiors (bathroom,bedroom,kitchen,living) from MSCOCO data
Build Bayes Machine learning models Decision Tree, Random Forest , Naïve Compare those.
Run test data through Clarify, Build Deep learning algorithms - SoftMax, CNN . Compare the accuracy between all the models. Extract VQA related to interiors and build VQA.

- **Data Source**

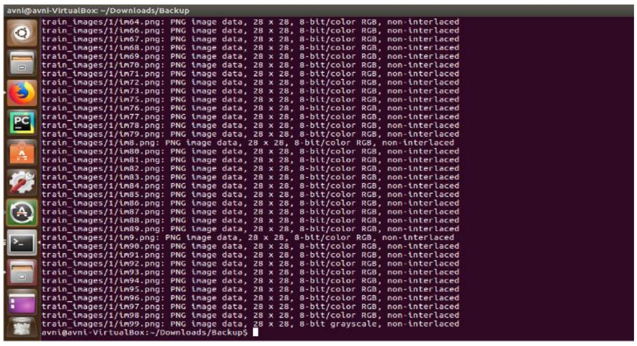
We have downloaded the dataset from http://mscoco.org/dataset/#download. COCO is a large-scale object detection and segmentation dataset.
The following table describes the data:

	Training Set	**Validation Set**	**Testing Set**
Images	82,783	40,504	81,434
Questions	2443,757	214,354	447,793
Answers	4,437,570	2,143,540	-

Segregate Interiors (bathroom, bedroom, kitchen, living) from MSCOCO data, 50 training images of each category and 5 test images.

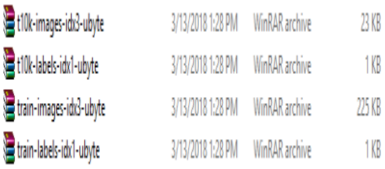
In order to perform the SoftMax and CNN on our project dataset, we had to perform the following pre-processing steps:

- Resize all the training and test images to 28x28 pixels.



```
avni@avni-VirtualBox: ~/Downloads/Backup
train_images/1/im64.png: PNG image data, 28 x 28, 8-bit/color RGB, non-interlaced
train_images/1/im66.png: PNG image data, 28 x 28, 8-bit/color RGB, non-interlaced
train_images/1/im67.png: PNG image data, 28 x 28, 8-bit/color RGB, non-interlaced
train_images/1/im68.png: PNG image data, 28 x 28, 8-bit/color RGB, non-interlaced
train_images/1/im69.png: PNG image data, 28 x 28, 8-bit/color RGB, non-interlaced
train_images/1/im70.png: PNG image data, 28 x 28, 8-bit/color RGB, non-interlaced
train_images/1/im71.png: PNG image data, 28 x 28, 8-bit/color RGB, non-interlaced
train_images/1/im72.png: PNG image data, 28 x 28, 8-bit/color RGB, non-interlaced
train_images/1/im73.png: PNG image data, 28 x 28, 8-bit/color RGB, non-interlaced
train_images/1/im75.png: PNG image data, 28 x 28, 8-bit/color RGB, non-interlaced
train_images/1/im76.png: PNG image data, 28 x 28, 8-bit/color RGB, non-interlaced
train_images/1/im77.png: PNG image data, 28 x 28, 8-bit/color RGB, non-interlaced
train_images/1/im78.png: PNG image data, 28 x 28, 8-bit/color RGB, non-interlaced
train_images/1/im79.png: PNG image data, 28 x 28, 8-bit/color RGB, non-interlaced
train_images/1/im8.png: PNG image data, 28 x 28, 8-bit/color RGB, non-interlaced
train_images/1/im80.png: PNG image data, 28 x 28, 8-bit/color RGB, non-interlaced
train_images/1/im81.png: PNG image data, 28 x 28, 8-bit/color RGB, non-interlaced
train_images/1/im82.png: PNG image data, 28 x 28, 8-bit/color RGB, non-interlaced
train_images/1/im83.png: PNG image data, 28 x 28, 8-bit/color RGB, non-interlaced
train_images/1/im84.png: PNG image data, 28 x 28, 8-bit/color RGB, non-interlaced
train_images/1/im85.png: PNG image data, 28 x 28, 8-bit/color RGB, non-interlaced
train_images/1/im86.png: PNG image data, 28 x 28, 8-bit/color RGB, non-interlaced
train_images/1/im87.png: PNG image data, 28 x 28, 8-bit/color RGB, non-interlaced
train_images/1/im88.png: PNG image data, 28 x 28, 8-bit/color RGB, non-interlaced
train_images/1/im89.png: PNG image data, 28 x 28, 8-bit/color RGB, non-interlaced
train_images/1/im9.png: PNG image data, 28 x 28, 8-bit/color RGB, non-interlaced
train_images/1/im90.png: PNG image data, 28 x 28, 8-bit/color RGB, non-interlaced
train_images/1/im91.png: PNG image data, 28 x 28, 8-bit/color RGB, non-interlaced
train_images/1/im92.png: PNG image data, 28 x 28, 8-bit/color RGB, non-interlaced
train_images/1/im93.png: PNG image data, 28 x 28, 8-bit/color RGB, non-interlaced
train_images/1/im94.png: PNG image data, 28 x 28, 8-bit/color RGB, non-interlaced
train_images/1/im95.png: PNG image data, 28 x 28, 8-bit/color RGB, non-interlaced
train_images/1/im96.png: PNG image data, 28 x 28, 8-bit/color RGB, non-interlaced
train_images/1/im97.png: PNG image data, 28 x 28, 8-bit/color RGB, non-interlaced
train_images/1/im98.png: PNG image data, 28 x 28, 8-bit/color RGB, non-interlaced
train_images/1/im99.png: PNG image data, 28 x 28, 8-bit grayscale, non-interlaced
avni@avni-VirtualBox:~/Downloads/Backup$ 
```

- Change the image color from RGB to black and white.
- Convert the resized images to MNIST format



t10k-images-idx3-ubyte	3/13/2018 1:28 PM	WinRAR archive	23 KB
t10k-labels-idx1-ubyte	3/13/2018 1:28 PM	WinRAR archive	1 KB
train-images-idx3-ubyte	3/13/2018 1:28 PM	WinRAR archive	225 KB
train-labels-idx1-ubyte	3/13/2018 1:28 PM	WinRAR archive	1 KB

- **Analytical Tools**

We have used the following tools for this increment:

1. Shallow learning – Decision Tree, Naïve Bayes ,Random Forest Model
2. Clarify
3. Deep Learning – SoftMax
4. Deep Learning - CNN

- **Analytical Task**

Build Decision Tree, Naïve Bayes, Random Forest models. Out of all Random Forest got good accuracy.
Object detection using Random Forest
Found accuracy and object detection using clarify
Convert data set to ubyte
Found accuracy using SoftMax and CNN

- **Expected Inputs/Outputs**

Input: Images belonging to one of the four Interiors (bathroom, bedroom, kitchen, living) .For each category we had 50 images for training and 10 images for testing.

Expected Output:
Find the accuracy using Random Forest, Clarify, SoftMax and CNN compare the results.
Object detection using Random Forest.

1.3 Related Work

- **Open Source Projects**

There have been several papers that are working on tasks of image tagging, image captioning and text-based Q&A. A team from MIT has worked on ‘Simple Baseline for Visual Question Answering’.

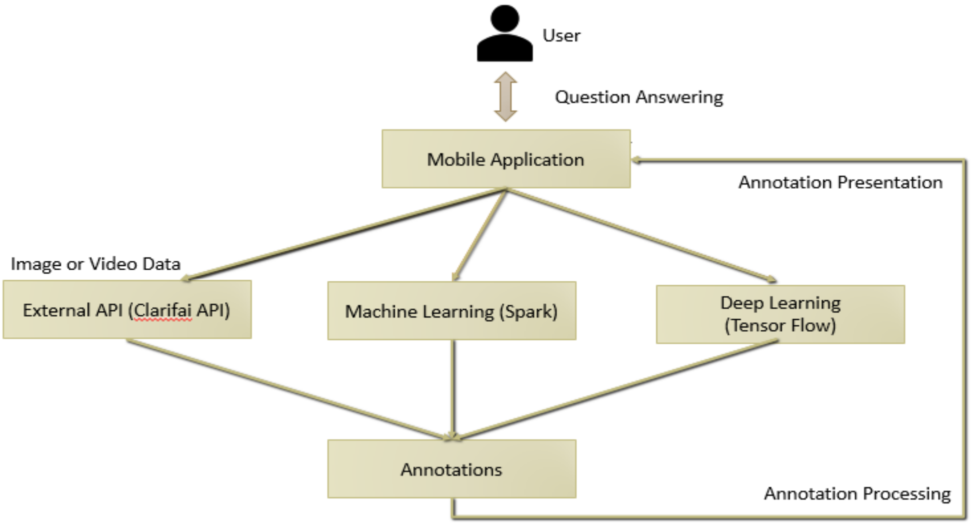
Show and Tell: Lessons learned from the 2015 MSCOCO Image Captioning Challenge

- **Literature Reviews**

We found the following scholarly papers related to Visual Question Answering topic:
[01] Learning to Answer Questions from Image Using Convolutional Neural Network
[02] An attention based convolutional neural network for visual question answering
[03] Exploring Models and Data for Image Question Answering

2.1 Application Specification & Implementation

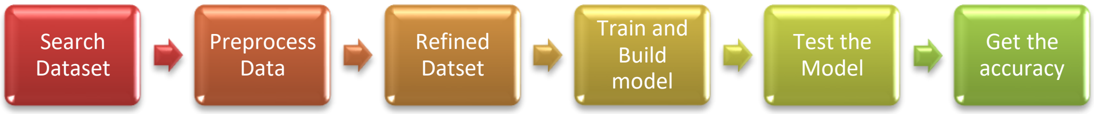
- **System Specification**
  - **Software Architecture**



The above figure represents the software architecture of this project. There will be a user interface in form of a web application where in the user can select image and ask the natural language question.

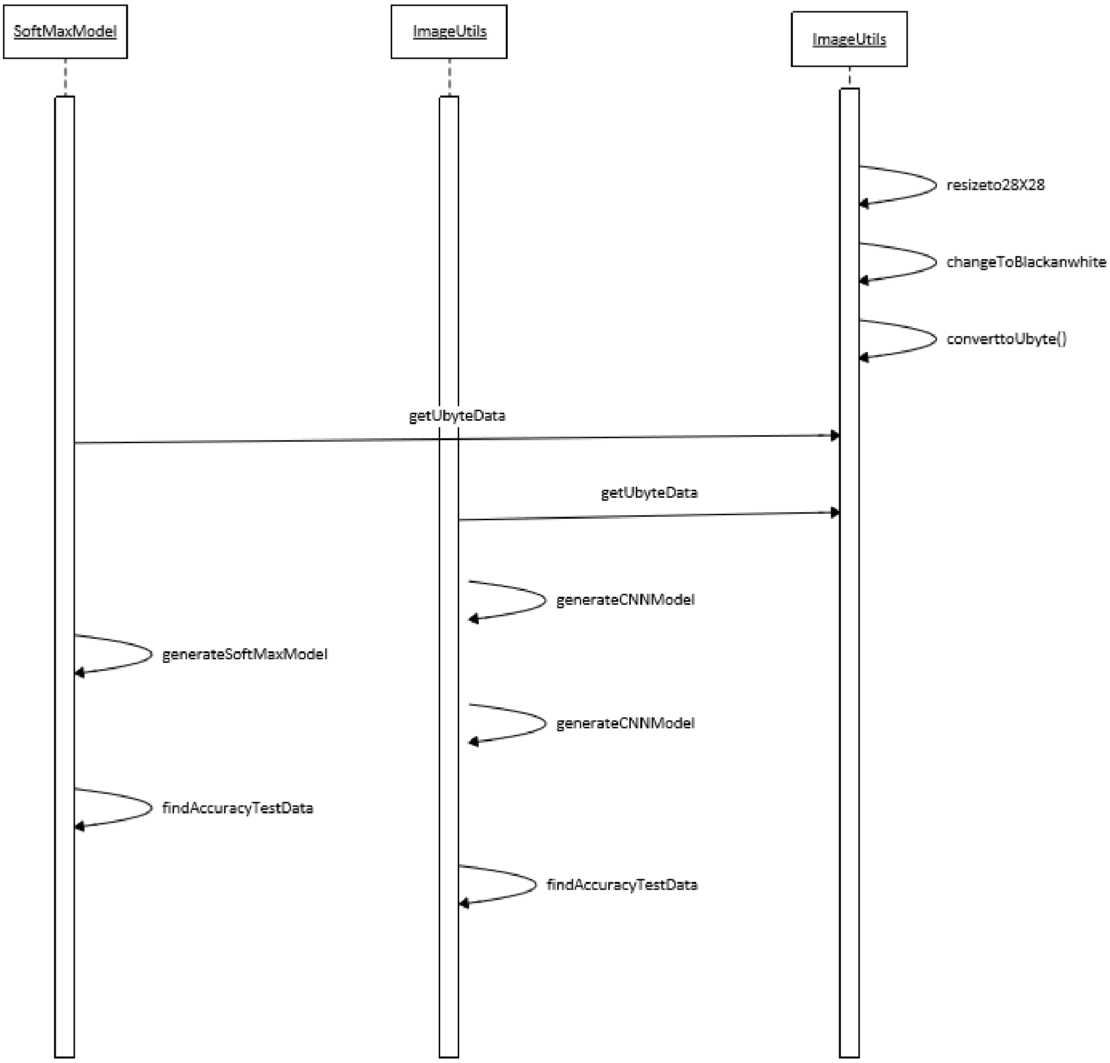
We have built following models Machine learning – Decision Tree, Naïve Baiyes, Random Forest Model , Deep Learning – SoftMax, CNN

- Features, Workflow, Technologies
    - Activity Diagram

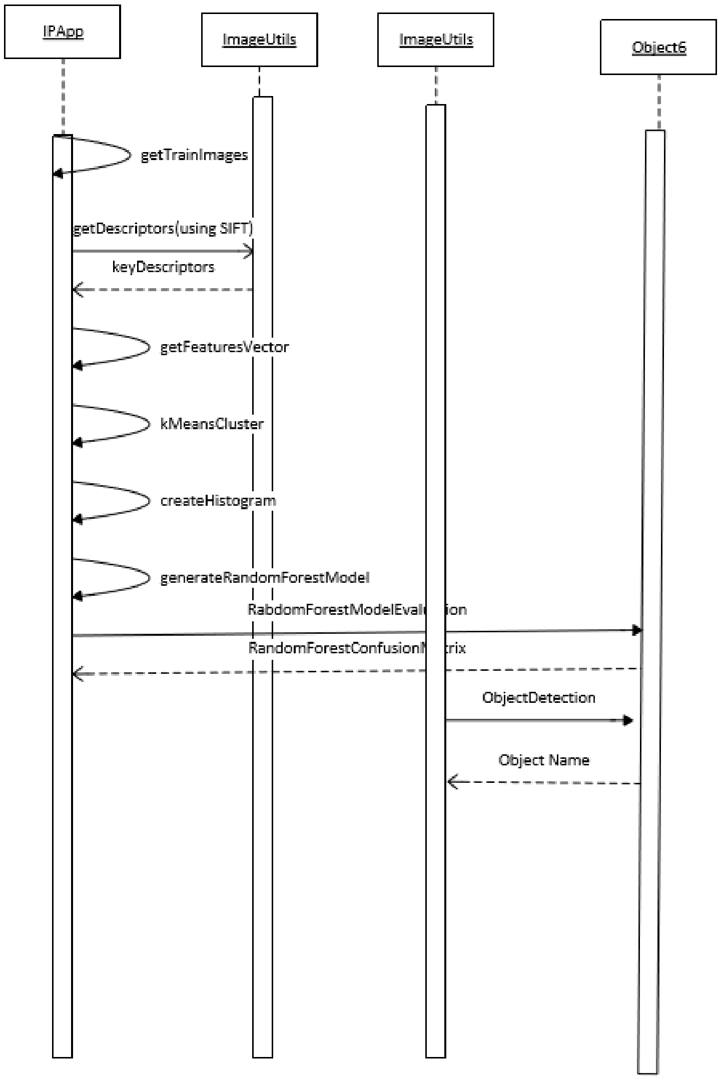


- Sequence Diagram

❖ Deep learning diagram



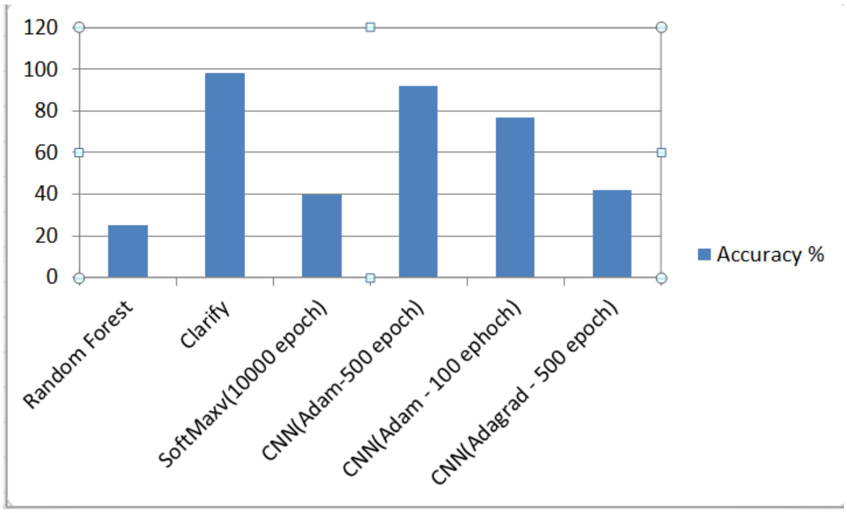
- Spark diagram



- Feature Specification

  - In this project increment, we have mainly focused on the Deep learning models (SoftMax, CNN) for our for interiors (bedroom, bathroom, kitchen, living) data.
  - Converted data to ubyte.
  - Compared accuracy with Random Forest Model, Clarify, SoftMax, CNN with different number of epoch and CNN AdamOptimizer, AdagradOptimizer

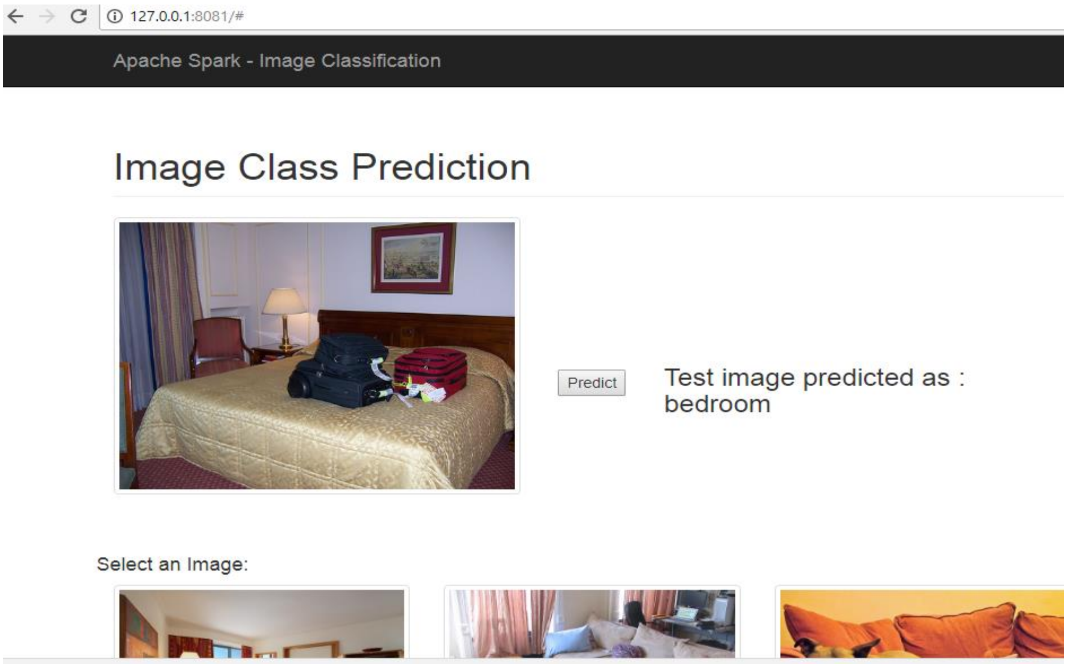
- Accuracy Matrix



Accuracy Comparison Matrix for MS COCOData for interiors (bedroom, bathroom, kitchen, living)

- Object Detection

Spark API Correct object Detection as Bedroom



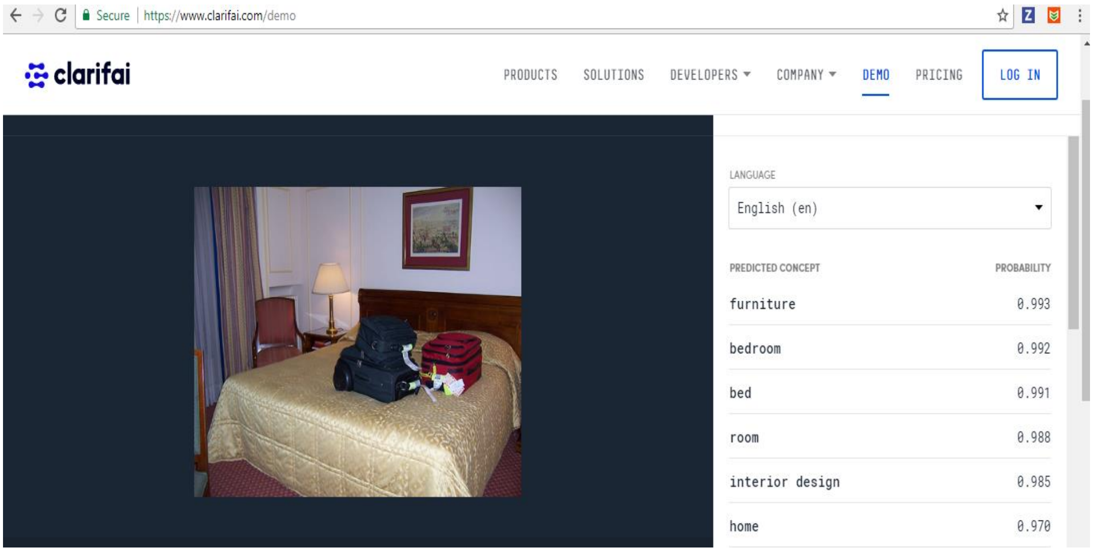
Apache Spark - Image Classification

Image Class Prediction

Test image predicted as : bedroom

Select an Image:

Clarify API



LANGUAGE

English (en)

PREDICTED CONCEPT	PROBABILITY
furniture	0.993
bedroom	0.992
bed	0.991
room	0.988
interior design	0.985
home	0.970

Spark API wrong object detection

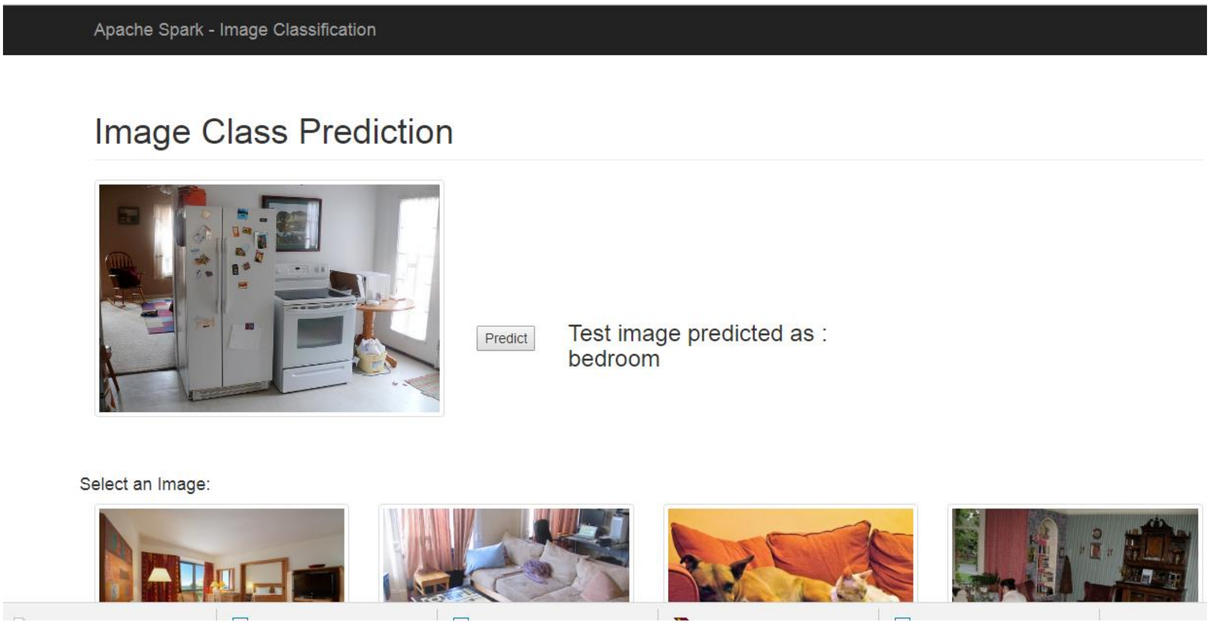
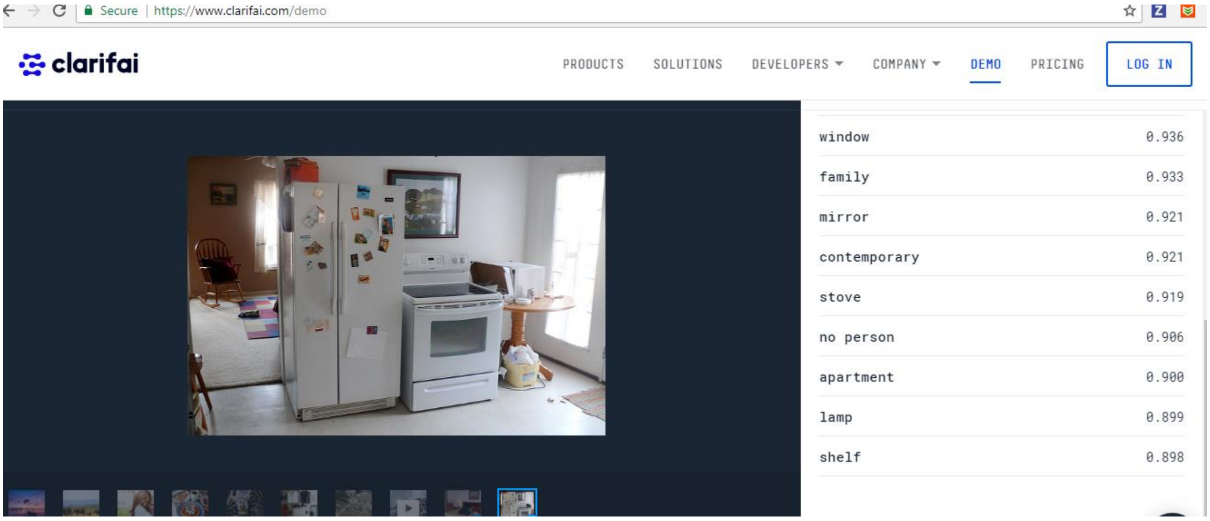


Image Class Prediction

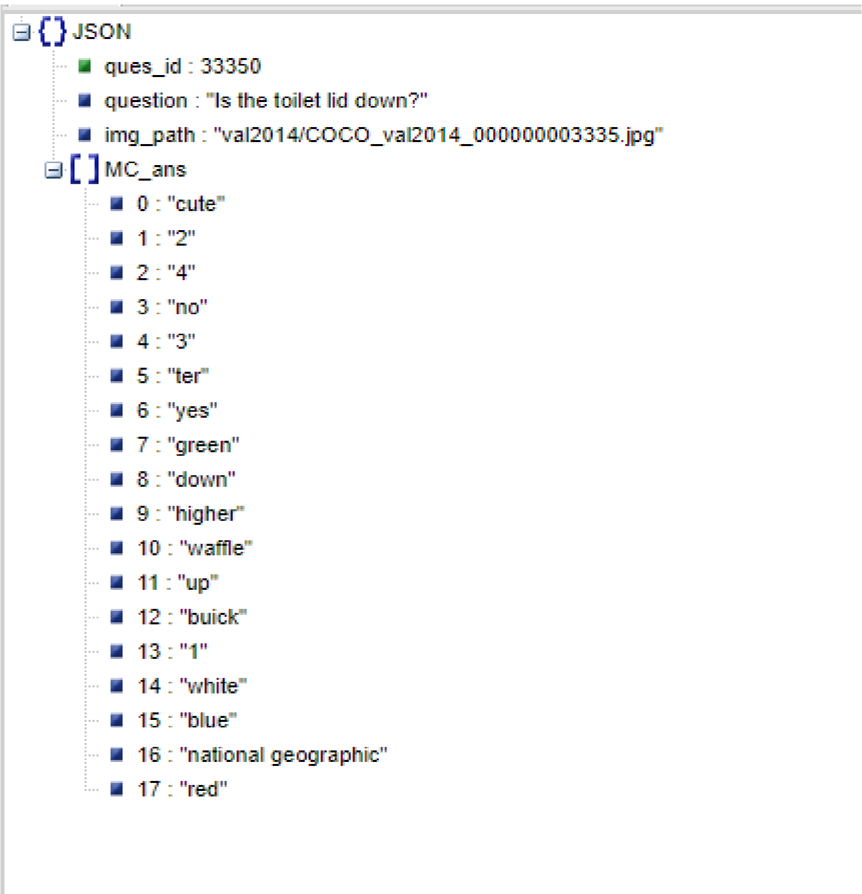
Test image predicted as : bedroom

Clarify no classification as Kitchen



window	0.936
family	0.933
mirror	0.921
contemporary	0.921
stove	0.919
no person	0.906
apartment	0.900
lamp	0.899
shelf	0.898

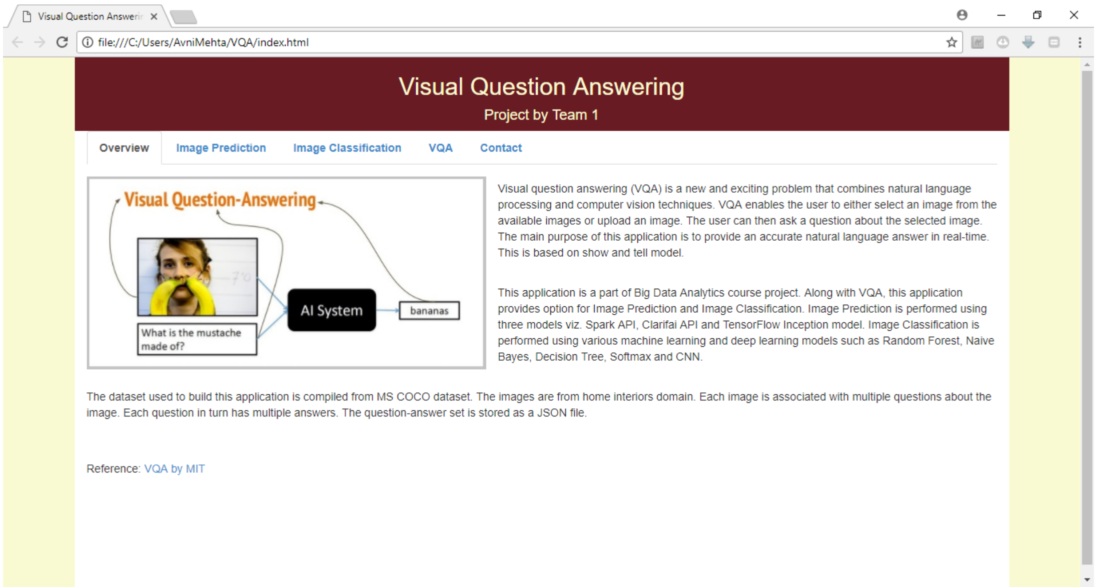
- QA question Json format



```
{ } JSON
    ques_id : 33350
    question : "Is the toilet lid down?"
    img_path : "val2014/COCO_val2014_000000003335.jpg"
  [ ] MC_ans
        0 : "cute"
        1 : "2"
        2 : "4"
        3 : "no"
        4 : "3"
        5 : "ter"
        6 : "yes"
        7 : "green"
        8 : "down"
        9 : "higher"
        10 : "waffle"
        11 : "up"
        12 : "buick"
        13 : "1"
        14 : "white"
        15 : "blue"
        16 : "national geographic"
        17 : "red"
```

- Web application

❖ Home


Visual Question Answerin
file:///C:/Users/AvniMehta/VQA/index.html
Overview
Image Prediction
Image Classification
VQA
Contact
Visual Question-Answering
What is the mustache made of?
AI System
bananas

Visual Question Answering

Project by Team 1

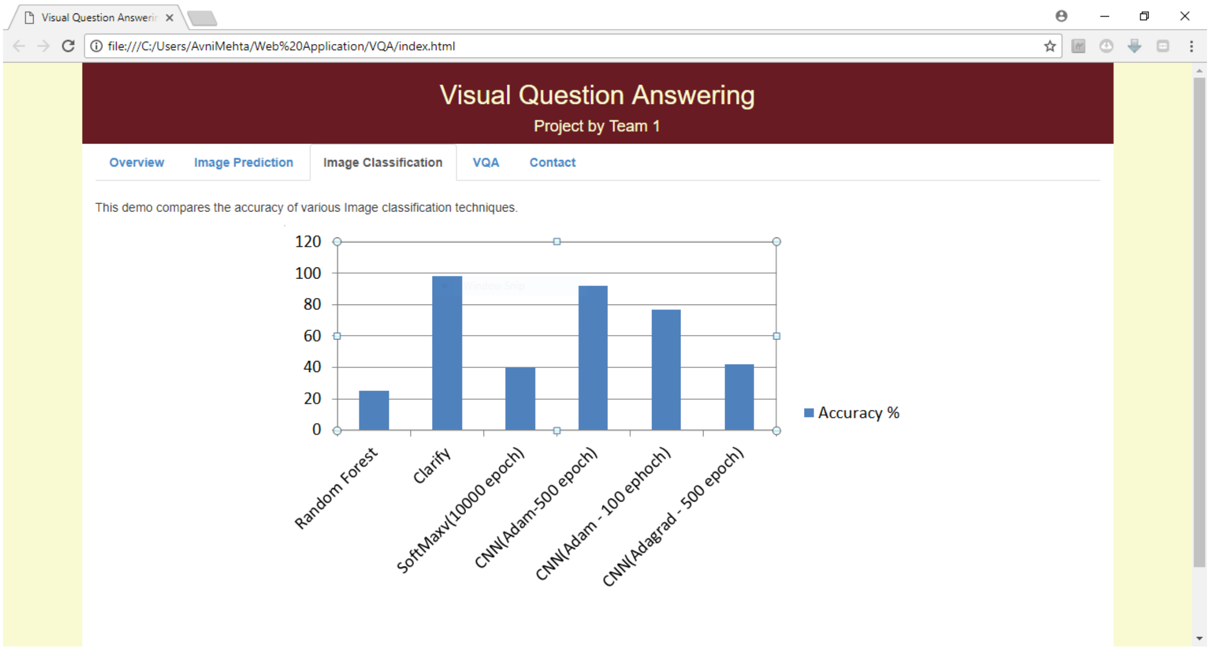
Visual question answering (VQA) is a new and exciting problem that combines natural language processing and computer vision techniques. VQA enables the user to either select an image from the available images or upload an image. The user can then ask a question about the selected image. The main purpose of this application is to provide an accurate natural language answer in real-time. This is based on show and tell model.

This application is a part of Big Data Analytics course project. Along with VQA, this application provides option for Image Prediction and Image Classification. Image Prediction is performed using three models viz. Spark API, Clarifai API and TensorFlow Inception model. Image Classification is performed using various machine learning and deep learning models such as Random Forest, Naive Bayes, Decision Tree, Softmax and CNN.

The dataset used to build this application is compiled from MS COCO dataset. The images are from home interiors domain. Each image is associated with multiple questions about the image. Each question in turn has multiple answers. The question-answer set is stored as a JSON file.

Reference: VQA by MIT

❖ Image Classification

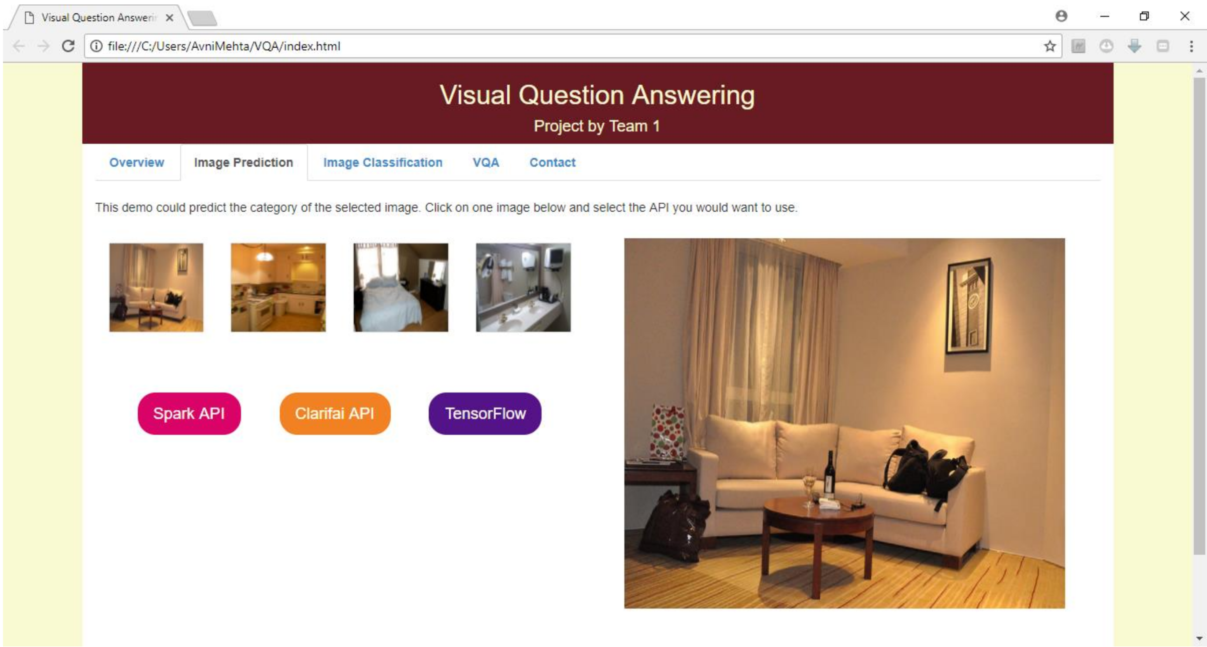


Visual Question Answering

Project by Team 1

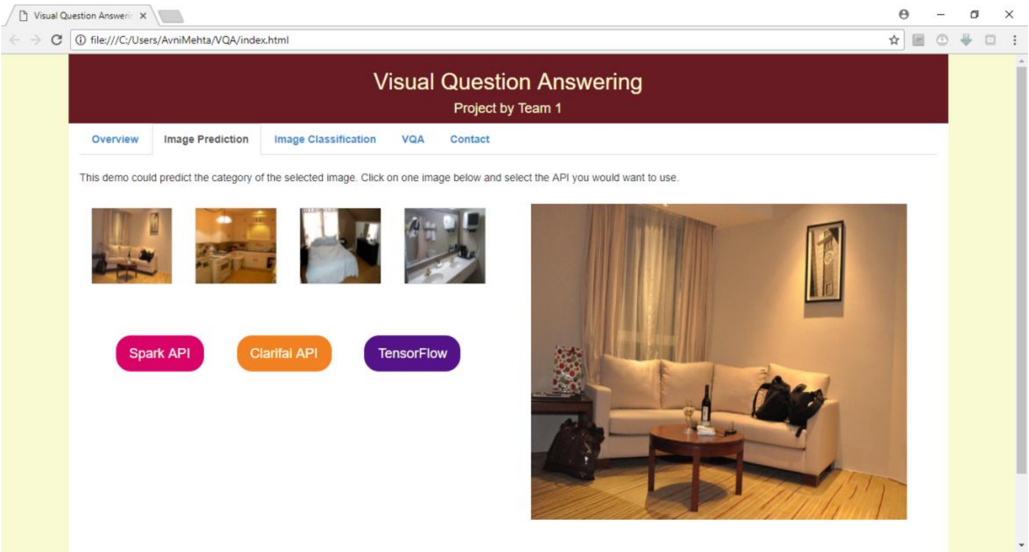
This demo compares the accuracy of various Image classification techniques.

❖ Image Prediction



This demo could predict the category of the selected image. Click on one image below and select the API you would want to use.

❖ Visual Question Answering



This demo could predict the category of the selected image. Click on one image below and select the API you would want to use.

- **Project Management**
  - **Timelines:**
    Increment 3: 4/23/2018
    Final Project Video/PPT: 4/30/2018
    Project Demo: 5/3/2018
    Final Project Package: 5/7/2018

  - **Team Members:**
    Santosh Yadav Jada
    Harshil Lavjibhai Patel
    Trinadha Rajeswari Muppala
    Avni Mehta

  - **Implementation Status:**
    - **Work completed:**
      We have worked on a MS COCO dataset and picked up four different categories of pictures Interiors (bathroom, bedroom, kitchen, living). We built our model, trained and tested the initial data.
      - Responsibility:
        - Search Datasets, Preprocessing (Santosh, Avni, Raji)
        - Run on the machine, Refinement process, Model (Avni, Raji, Harshil)
        - Web application (Avni)
        - Build MIT VQA model do reverse engineering (Harshil)
        - Ideas (Avni, Santosh, Raji)
        - Documentation (Avni, Santosh, Harshil)
      - Time taken: Good amount of hours for searching the data sets and proposing ideas but for pre-processing & documentation (26 hours approx.)
      - Contributions: As a team we worked together and brought the results, with 25% contribution from everyone.

    - **Work to be completed:**
      We are working towards getting a specific domain in the MS COCO set and design a system which could give us good accuracy in answering the questions we pose to the machine regarding the domain. This includes making of Mobile application, integrating the built model to the mobile app and tuning the parameters.
      - Responsibility: We are complementing our skillset and working for the completion of the project.
      - Time required: We are estimating at least 200 hours to get the rudimentary application and some more hours to get it refined and user friendly.
      - Contributions: Team spirited individuals with 25% contribution from everyone.

    - **Issues/Concerns:**
      - We have extracted Question Answers from MS COCO question answers. Having difficult how to connect with image classification with Question answers