Visual Question Answering

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

Abstract

In the world of computer vision, there are many projects on visual question answering, but we put forth this novel methodology to get answers from images and also tried to incorporate a visual question answering system with the help of bounding box concept. we are taking MS COCO data which got 80,000 images with different domain. We could use this system to help visually impaired people to identify the objects in a given image. This is an application that enables the user to either select an image from the available images or upload an image. The user can ask a question about the selected image. The main purpose of this application is to provide a natural language answer in real-time.

1. Introduction

Image processing and computer vision has become popular over the last few years and research is growing exponentially. In this project we are using MS COCO data set and limiting our VQA model to interior (bathroom, bedroom, kitchen, living room). We already built models using Machine Learning and Deep learning algorithms like CNN. We are using bounding box to identify the object accurately. This is done when we use Mask R-CNN with python and tensorflow. We are creating a box by picking a smallest box that captures all the pixels of the mask as the bounding box.

2. 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 MS-COCO 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

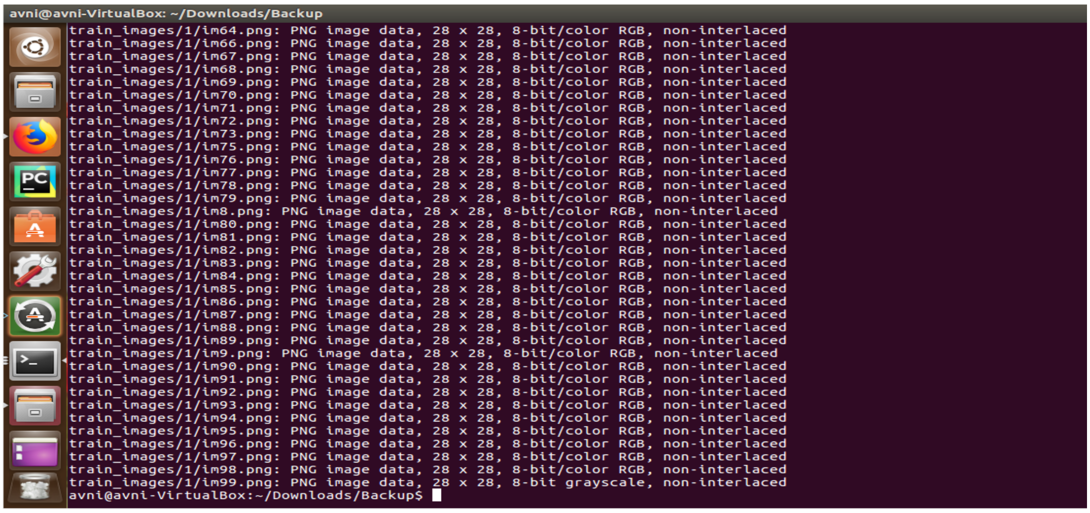
3. Approach

- Segregate Interiors (bathroom, bedroom, kitchen, living) from MSCOCO data.
- Build Bayes Machine learning models Decision Tree, Random Forest.
- Compare the results. 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.

- Implement bounding box by using mask R-CNN with python and tensorflow.

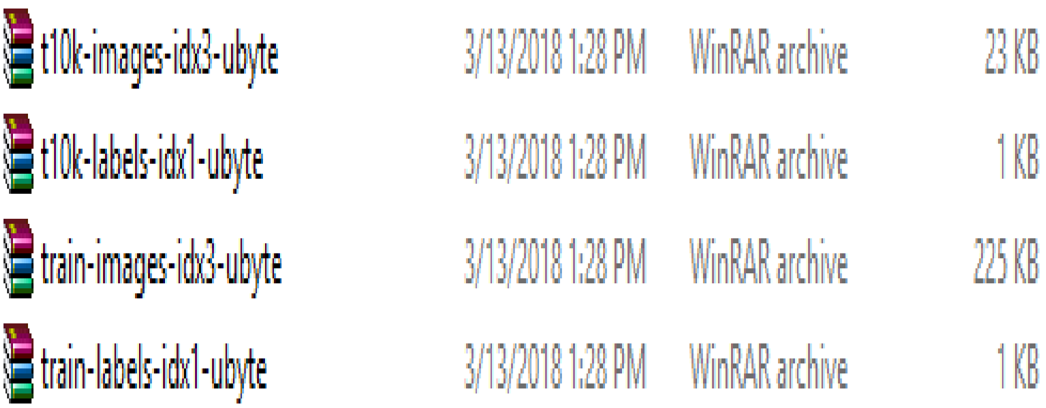
In order to perform the SoftMax and CNN on our project dataset, we had to perform the following pre-processing steps:

1. 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$ 
```

2. Change the image color from RGB to black and white.
3. 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 API, Spark API
  3. Deep Learning – SoftMax
  4. Deep Learning – CNN + Mask R-CNN

- **Analytical Task**

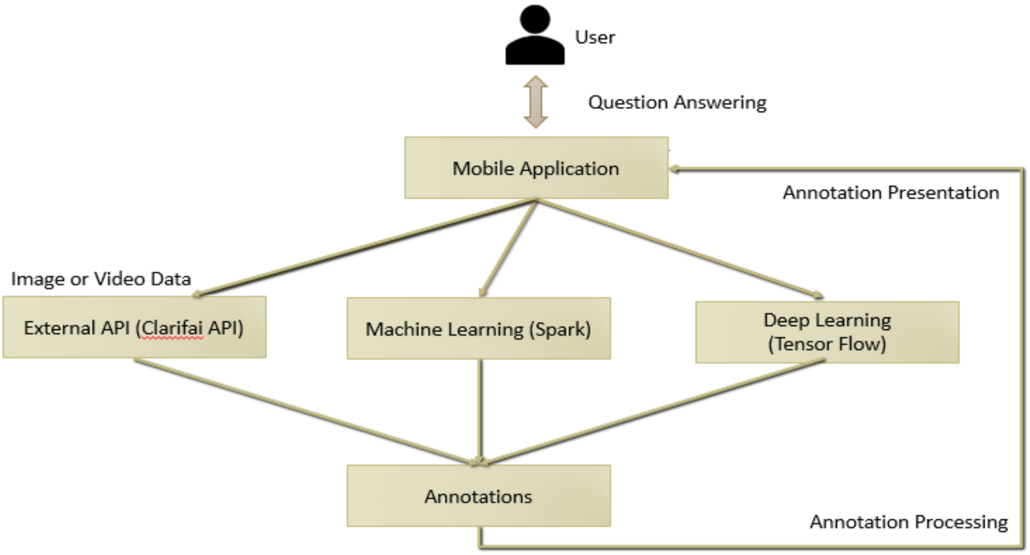
  1. Build Decision Tree, Naïve Bayes, Random Forest models. Out of all Random Forest got good accuracy.
  2. Object detection using Random Forest.
  3. Found accuracy and object detection using clarify.
  4. Convert data set to ubyte.
  5. Found accuracy using SoftMax and CNN.
  6. Implementation of bounding box using Mask R-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.
  Find out different objects using bounding box
  Answer the question asked by the user.

4. Application Specification & Implementation

System Specification

- **Software Architecture**

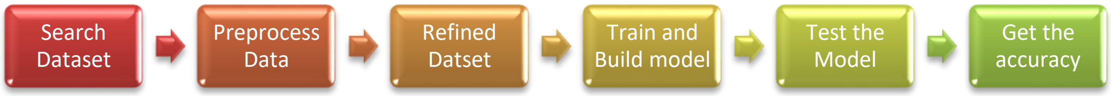


The above figure represents the software architecture of this project. There will be a user interface in form of a web application where in the user can select image and ask the natural language question.

We have built following models Machine learning – Decision Tree, Naïve Baiyes, Random Forest Model, Deep Learning – SoftMax, CNN and also Mask R-CNN.

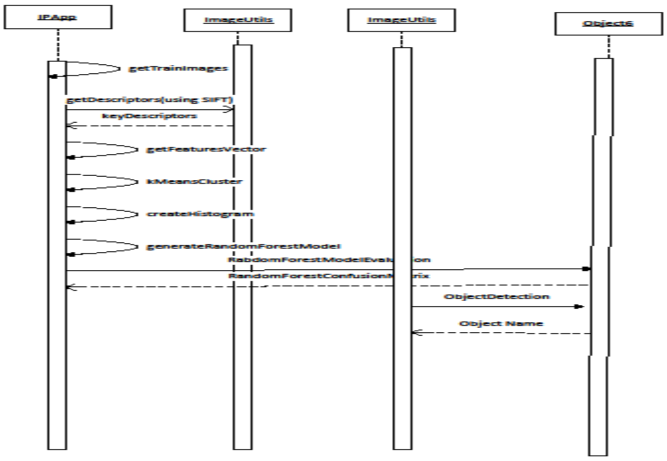
- **Features, Workflow, Technologies**

1. Activity Diagram
   The activity diagram tells the process of how searching the dataset, preprocessing stage, refining the dataset, training the model and then building the model, testing the model and finally tuning the parameters to get the accuracy.

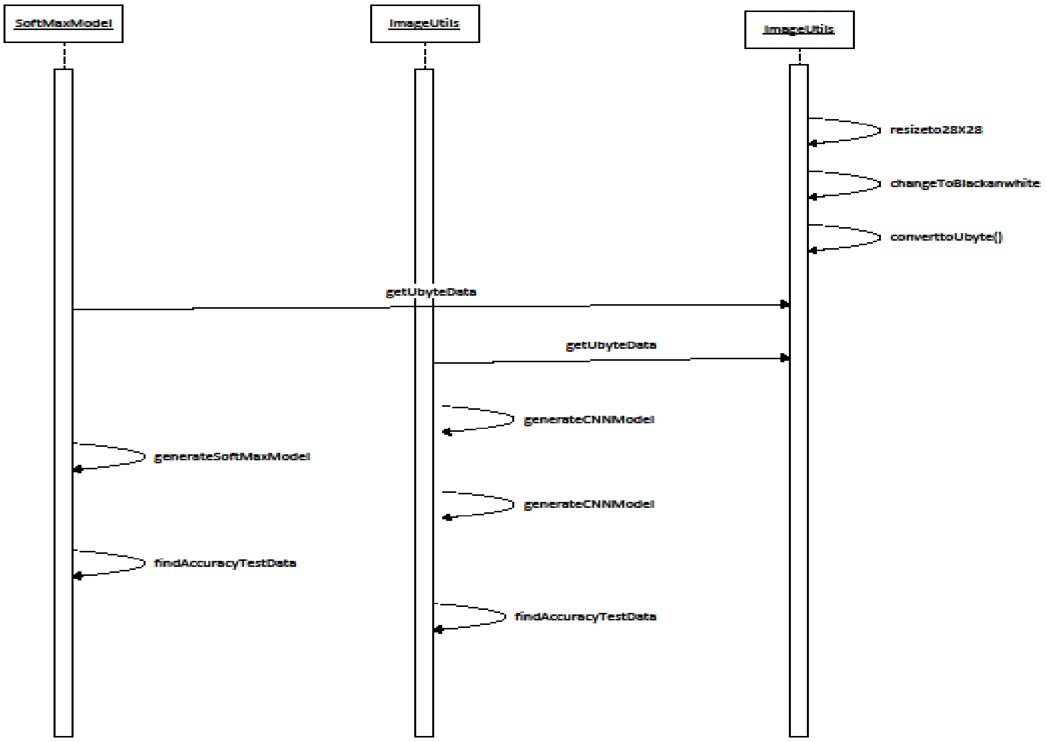


2. Sequence Diagram:

   This is the sequence diagram when we use Spark :



This is the sequence diagram when we use deep learning:

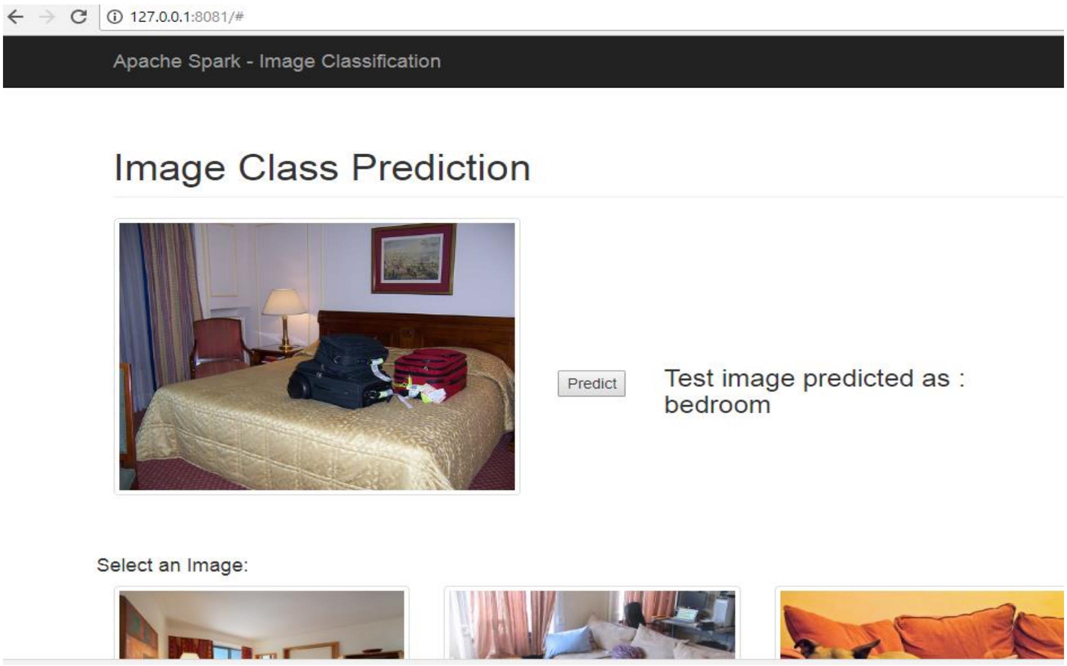


- ## Feature Specification

1. In this project increment, we have mainly focused on the Deep learning models (SoftMax, CNN) for our for interiors (bedroom, bathroom, kitchen, living) data.
2. Converted data to ubyte.
3. Compared accuracy with Random Forest Model, Clarify, SoftMax, CNN with different number of epoch and CNN AdamOptimizer, AdagradOptimizer
4. Usage of bounding box using Mask R-CNN. It is done by using python and tensor flow. This model generates bounding boxes for objects in the image.
5. The weight histograms generated would be helpful in the identification. We started of with anchor sorting and filtering, then refinement is applied on the bounding box. Then the images are scaled.
6. It also shows the percentage of the prediction.

- **Object Detection**

Spark API Correct object Detection as Bedroom



127.0.0.1:8081/#

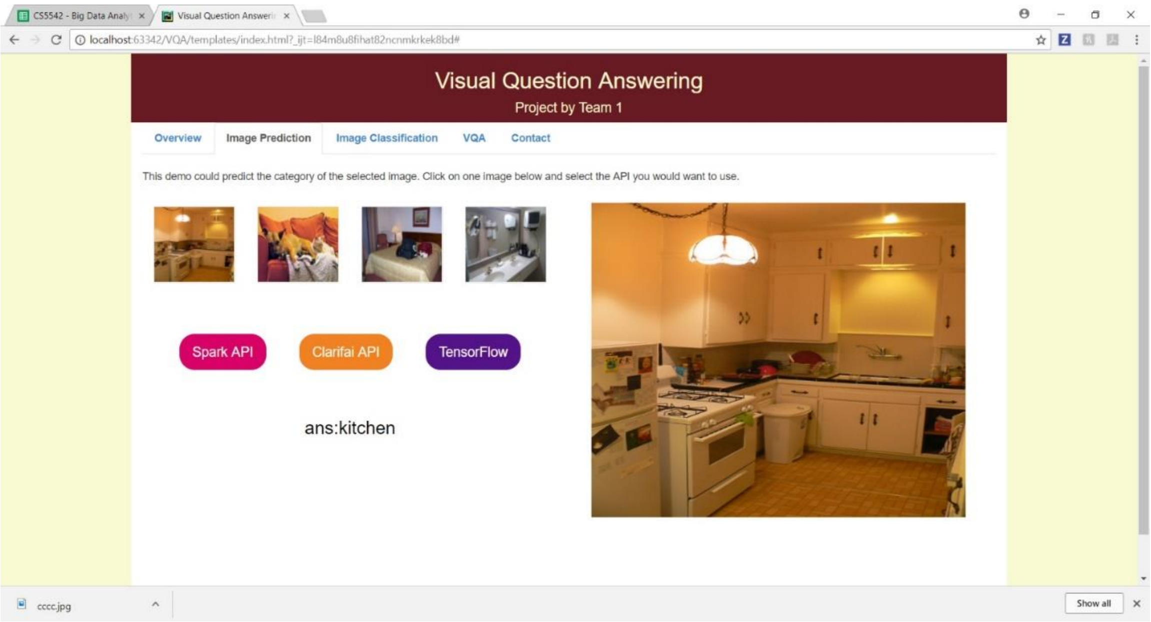
Apache Spark - Image Classification

Image Class Prediction

Test image predicted as : bedroom

Select an Image:

- **Identification using Clarifai:**



localhost:63342/VQA/templates/index.html?_ijt=l84m8u8fihat82ncnmkrkek8bd#

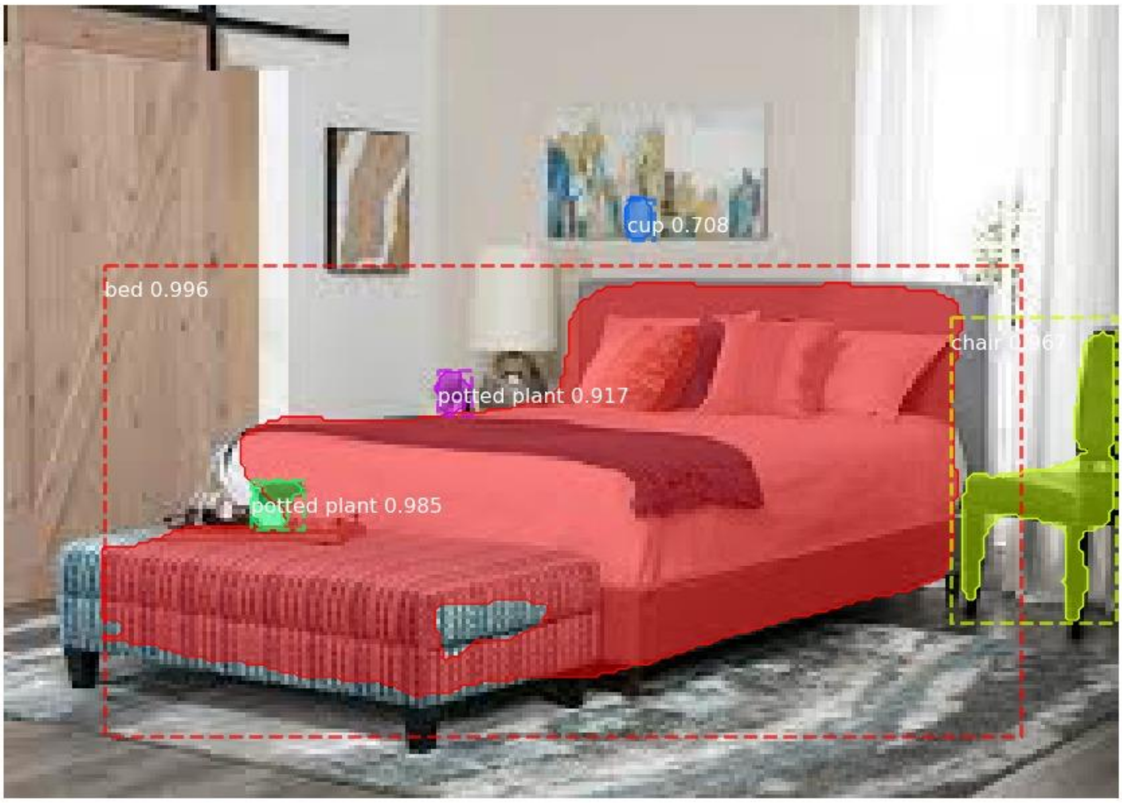
Visual Question Answering

Project by Team 1

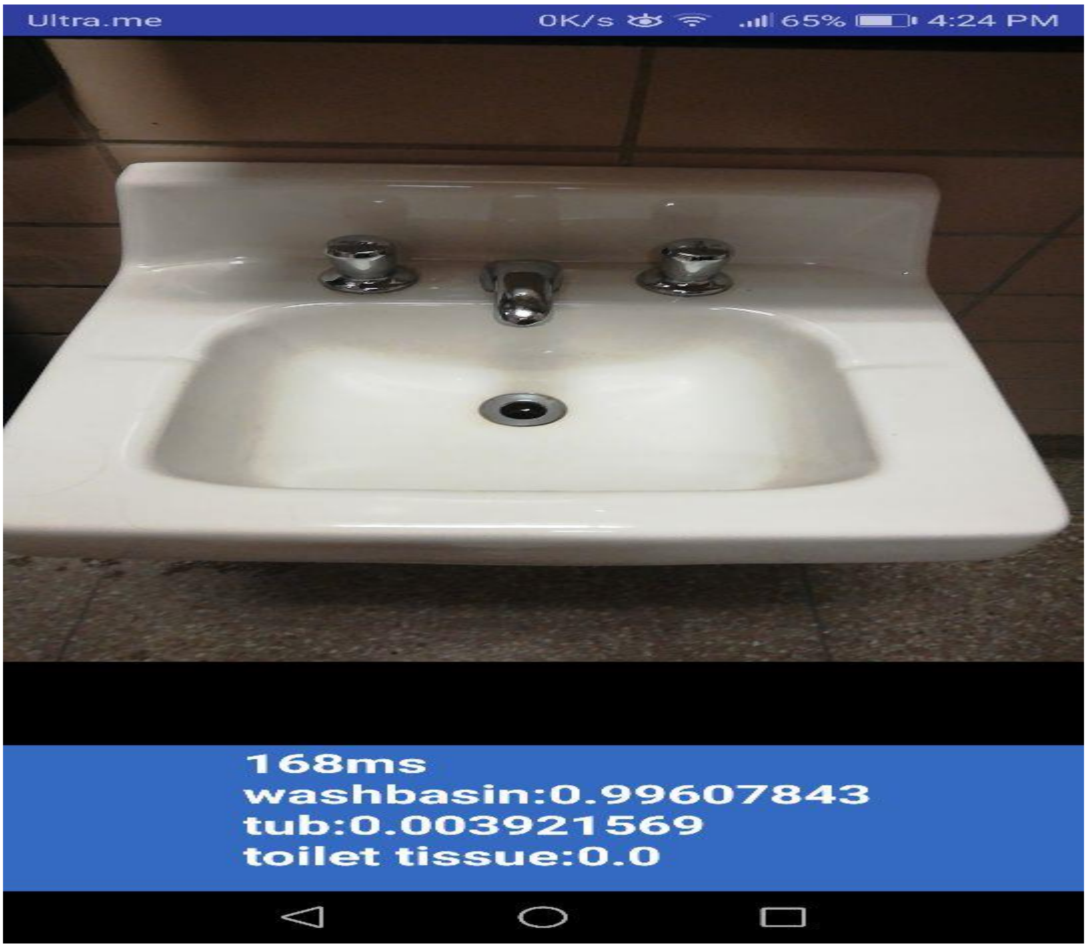
This demo could predict the category of the selected image. Click on one image below and select the API you would want to use.

ans:kitchen

- **Bounding box:**



- **Tensor Flow Lite:**
  We have implemented the object prediction on tensor flow lite for our category (interiors)

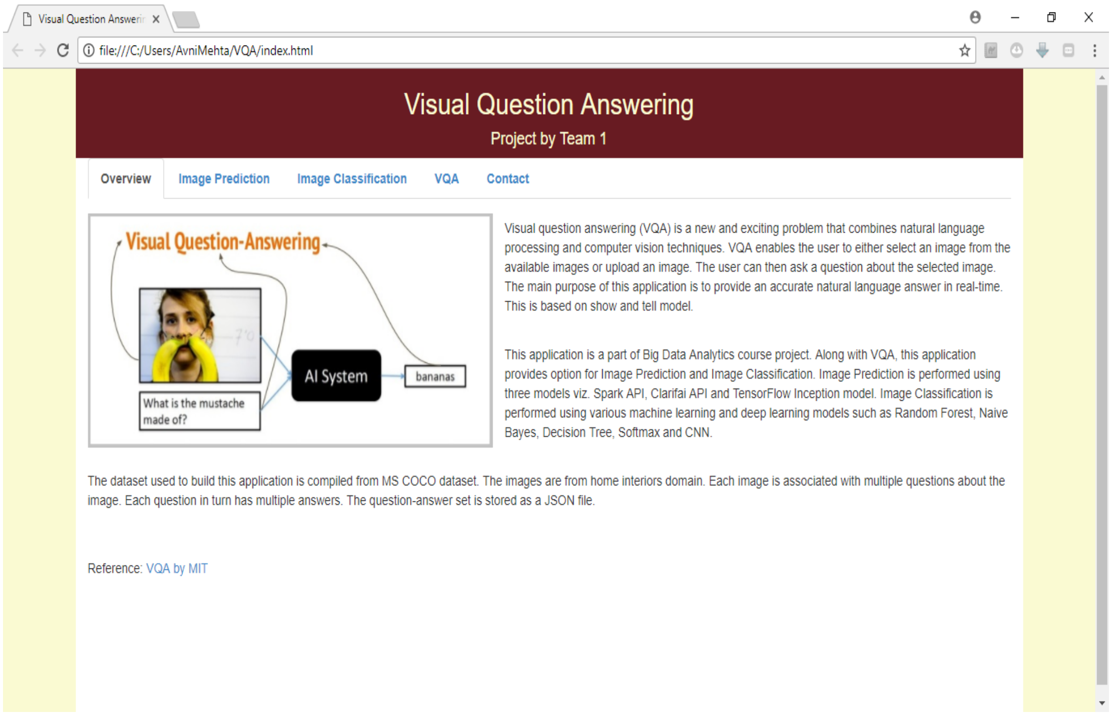


```
168ms
washbasin:0.99607843
tub:0.003921569
toilet tissue:0.0
```

- **Web application**

  1. The home page of the website looks this where any user can navigate and see different tabs associated. The Main Page talks about the project and the other tabs are self-explanatory.
  2. The image classification tab talks about the different results obtained.
  3. The image prediction tab shows the predicted answer for the pic which is chosen.
  4. In image prediction we have three methods to do it namely Spark API, Clarifai API and Tensor Flow.
  5. Integration of the website using the python flask library.

Home



Visual Question Answeri
file:///C:/Users/AvniMehta/VQA/index.html

Visual Question Answering

Project by Team 1

Overview
Image Prediction
Image Classification
VQA
Contact
Visual Question-Answering
AI System
bananas
What is the mustache made of?

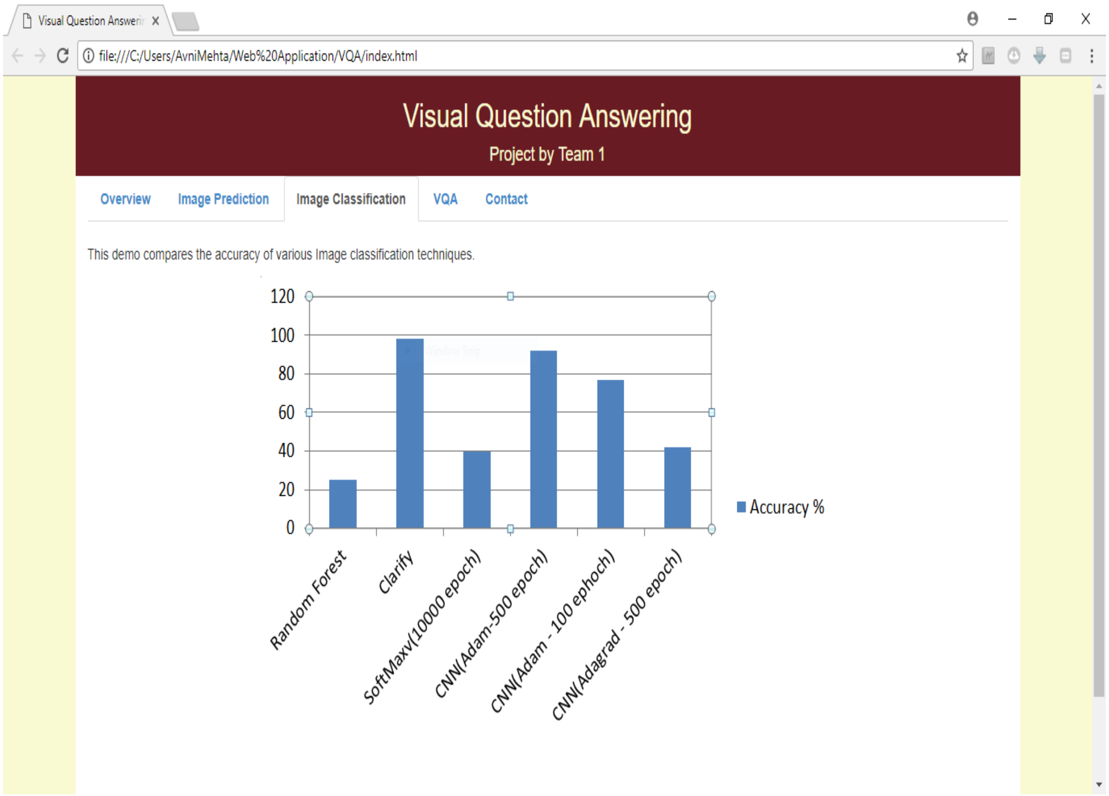
Visual question answering (VQA) is a new and exciting problem that combines natural language processing and computer vision techniques. VQA enables the user to either select an image from the available images or upload an image. The user can then ask a question about the selected image. The main purpose 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

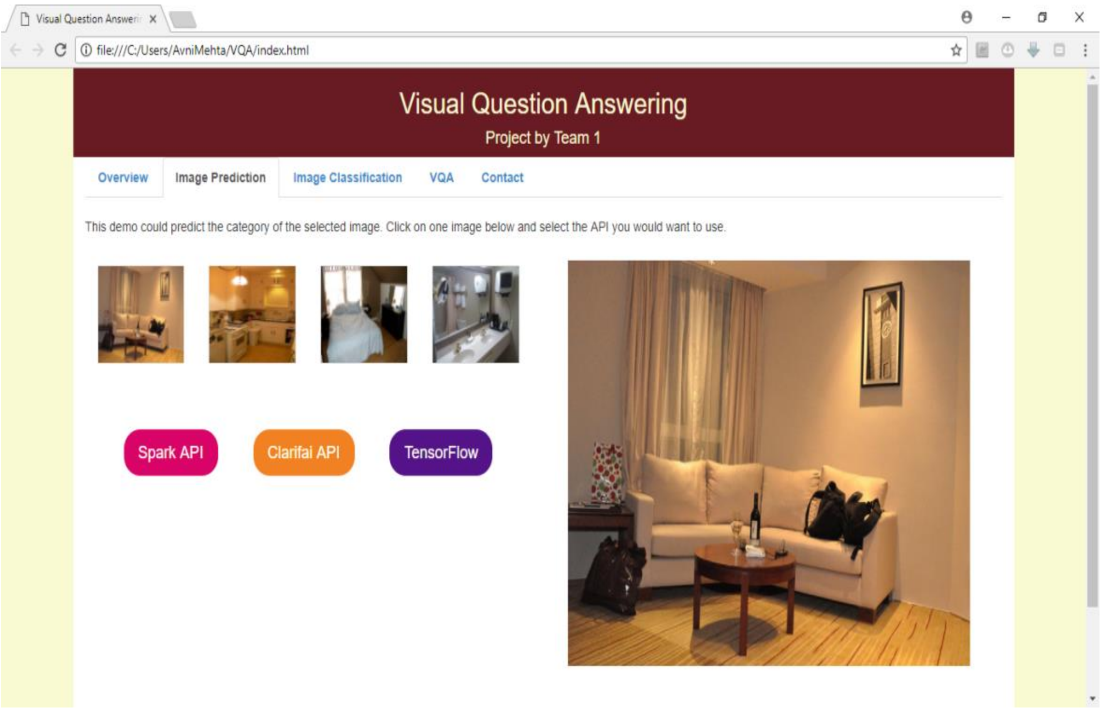


Visual Question Answering

Project by Team 1

This demo compares the accuracy of various Image classification techniques.

Visual Question Answering

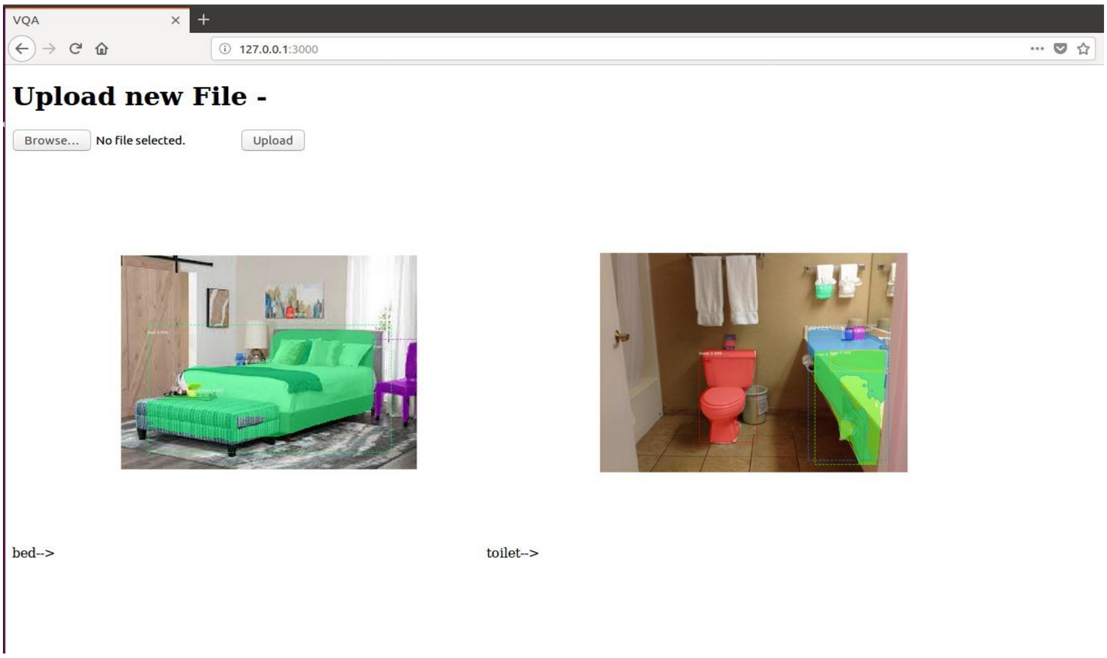


Visual Question Answering

Project by Team 1

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

Upload the image:



Upload new File -

No file selected.

bed-->

toilet-->

5. Results:

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 room) from MSCOCO data, 50 training images of each category and 5 test images.

6. Conclusion:

We created a system which gives yes/no answer for the question asked based on the image. This system takes a picture as input form the user through the website and processes it. When the user asks a question, then the system answers it in yes/no. Implementation of more complex answers and questions would take GPUs for processing. The outputs of different models have been compared and presented during this project. This project could work as a base for more complex projects.

7. Appendix:

1. Introduction
2. Related Work
3. Approach
4. Application specification & Implementation
5. Results
6. Conclusion
7. Appendix
8. References
9. Project Management

8. References:

1. https://github.com/matterport/Mask_RCNN

2. http://flask.pocoo.org/

3. http://www.visualqa.org/

4. https://cs.stanford.edu/people/karpathy/rcnn/

9. Project Management:

1. **Timelines:**
   Final Project Video/PPT: 4/30/2018
   Project Demo: 5/3/2018
   Final Project Package: 5/7/2018

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