
VQA – Visual Question Answering

Team# 1 | #5 Jada, Santosh | #11 Mehta, Avni | #14 Muppala, Raji | #19 Patel, Harshil

---

INCREMENT 1 REPORT

- **Project Objectives**
  - **Significance**

There are several projects on visual question answering; unlike our application, they are restricted with the questions from a fixed set and large domain. Our application involves open-ended, free-form questions and answers provided by humans and a specificity for the domain. Additionally, we are trying to work with speech-to-text so that it could be used by visually impaired.

  - **System Features**

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 could be extended as a mobile application. This is based on show and tell model.

- **Approach**

Machine learning algorithms (including Naïve Bayes, Random Forest and Decision Tree) were used to categorize images into categories and make predictions for the test images.

  - **Data Sources**

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	-

This is the total data, but we have taken four categories of data (people, animals, food and interiors) for this increment. Going forward, we would focus on a single domain.

  - **Analytical Tools**

We have used the following tools for this increment:
1. Clarifai API – For image classification
2. Spark – For libraries such as MLlib, KMeans, RandomForest, DecisionTree, RDD etc.

  - **Analytical Task**

The problem sets of the class has helped us frame the ideas to approach. There has been a hustle to get the dataset and also how to put it in the form of an application. We also ran our dataset using various classification approaches and compared the output.

  - **Expected Inputs/Outputs**

Input: Images belonging to one of the four categories (people, animal, food, interiors). For each category we had 50 images for training and 10 images for testing.
Expected Output:
The expected output is classification of the given image into the four categories along with model accuracy.

- **Algorithms**
  We have used Naïve Bayes, Random Forest and Decision Tree algorithms to build a model for our data.

❖ Related Work

- **Open Source Projects**

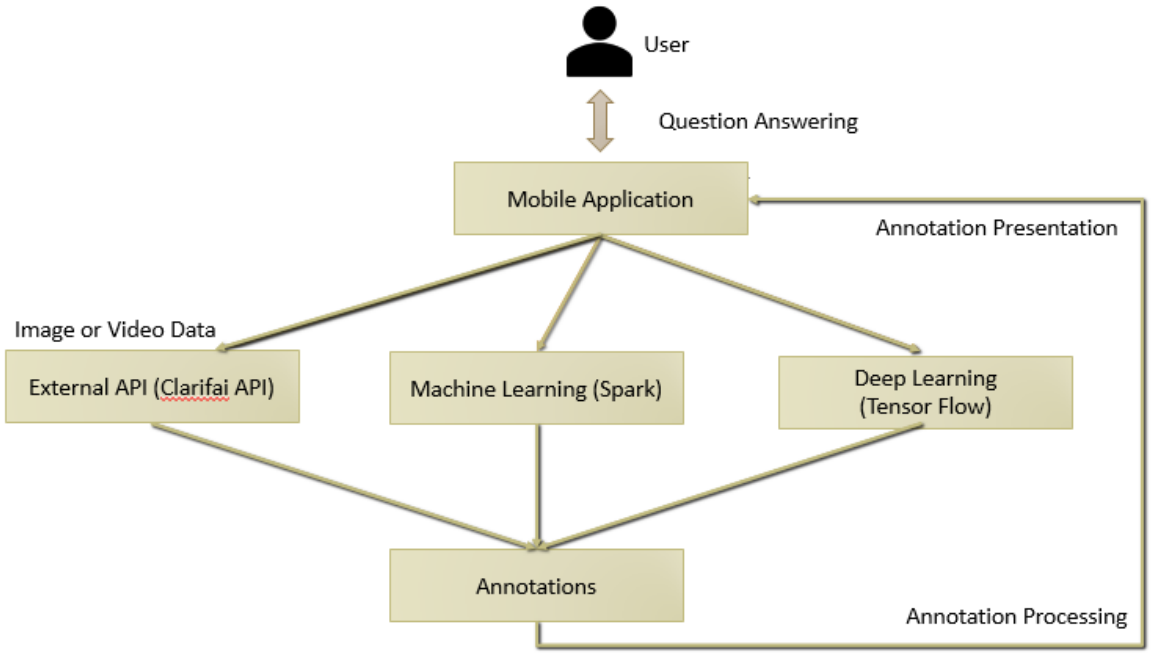
There have been several papers that are working on tasks of image tagging, image captioning and text-based Q&A. A team from MIT has worked on 'Simple Baseline for Visual Question Answering'.

- **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

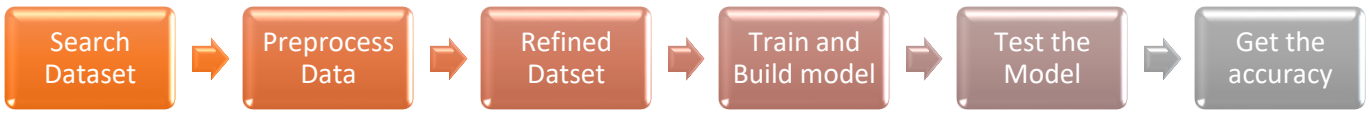
❖ Application Specification & Implementation

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

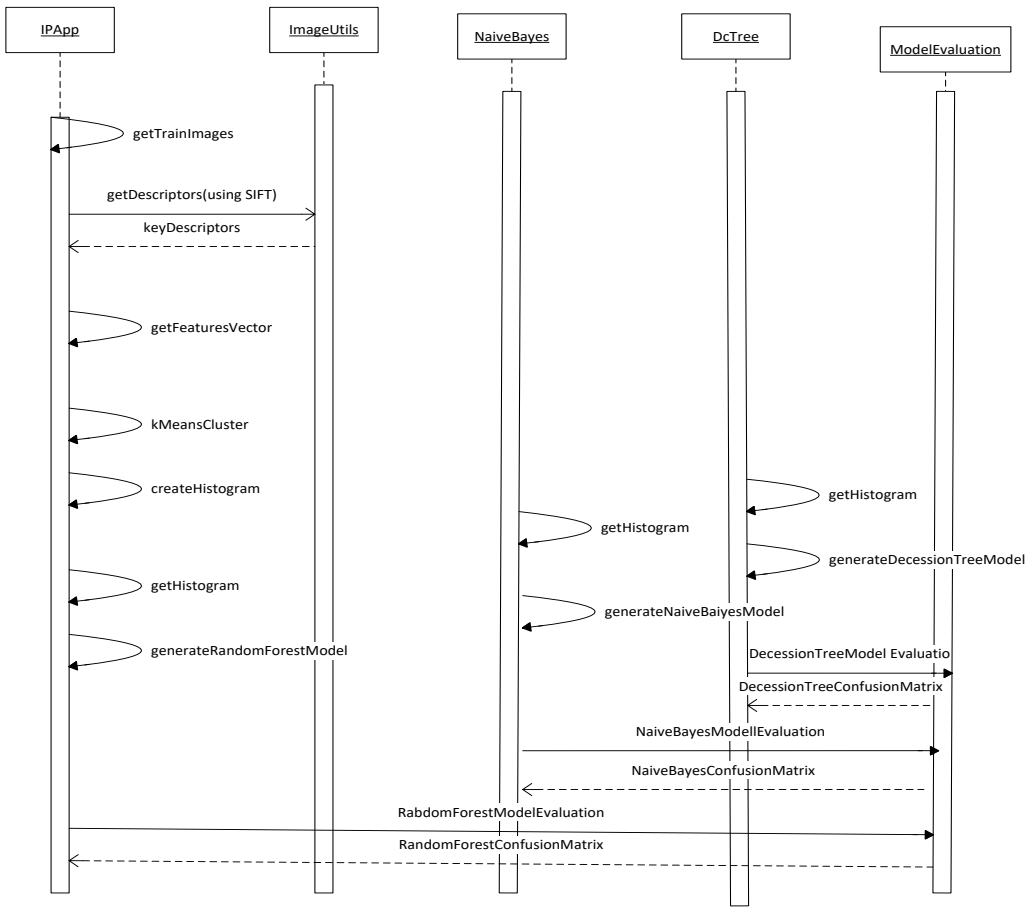


The above figure represents the software architecture of this project. There will be a user interface in form of a mobile application where in the user can select image and ask the natural language question. In the first part, we use Clarifai API and other machine learning libraries in Spark to build a classification model and predict the image. Next, we use TensorFlow to train our images through deep neural networks. This will consist of multiple layers of CNN and LSTM.

- Features, Workflow, Technologies
  - Activity Diagram



  - Sequence Diagram



- Feature Specification
  - In this project increment, we are mainly focusing on specific features of our domain so retrieve information using Image classification techniques (Scala).
  - Extracting main frames and generating features for further image analysis.
  - Using this information to process the new image and understanding is half way through the project.

- Operation Specification

Input	**Operation**	**Output**
Train images	SIFT.detectAndCompute	Features
Features	KMeans.train	Clusters
Clusters	Mat	Vocabulary, Histograms
Histograms	RandomForest.trainClassifier	Random Forest Model
Histograms	NaïveBayes.train	Naïve Bayes Model
Histograms	DecisionTree. trainClassifier	Decision Tree Model
Random Forest Model, Test Images	MultiClassMetrics MultiClassMetrics.confusionMatrix	Random Forest Confusion Matrix, Accuracy, Precision
Decision Tree Model, Test Images	MultiClassMetrics MultiClassMetrics.confusionMatrix	Decision Tree Model Confusion Matrix, Accuracy, Precision
Naïve Bayes Model, Test Images	MultiClassMetrics MultiClassMetrics.confusionMatrix	Naïve Bayes Confusion Matrix, Accuracy, Precision

- **Existing Applications/Services used:**
  1. Apache Spark
  2. Clarifai API
  3. OpenImg Library

- **Implementation of your application using Clarifai API:**
  We used Clarifai API to summarize a given video and OpenImg library to detect key frames. The below mentioned are the steps:
  1. In KeyFrameDetection.java, import required libraries for Spark, OpenImj and Clarifai API
  2. In public class KeyFrameDetection, for the given .mkv video, get all the frames from the video.
  3. For frames extraction, iterate over video frames and select the main frames in the video.
  4. Compare SIFT features with neighbouring images. When common features < certain threshold, shot the transition.
  5. Find all the main key points, collect them and output the results as mainframes.
  6. Connect to the Clarifai API server by using API key and access code for the token.
  7. Using this connection, access the mainframes file and scan the image in detail and predict information present in the image.
  8. Update the image with all the possible contents in each image. Output the image.

❖ **Project Management**

- **Timelines:**
  Increment 2: 3/19/2018
  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 (animals, interiors, people). We built our model, trained and tested the initial data.
    - Responsibility:
      - Search Datasets, Preprocessing (Santosh, Avni, Raji)
      - Run on the machine, Refinement process, Model (Raji, 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 complimenting 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:**
    - Getting better accuracy than previous models.