Hidden Hurdles

Thipparthi Manasa T,
School of Computing and Engineering
University of Missouri, Kansas City
mthwf@mail.umkc.edu

Gudibandi SaiJyothi,
School of Computing and Engineering
University of Missouri, Kansas City
sg583@mail.umkc.edu

Puthana Sujitha,
School of Computing and Engineering
University of Missouri, Kansas City
spb4b@mail.umkc.edu

Sri Harsha Kumar Raja Golla
School of Computing and Engineering
University of Missouri, Kansas City
sgnt8@mail.umkc.edu

Abstract: **The core purpose of our project is to classify the image using External API(Clarifai API),Machine Learning(Spark) and Deep Learning(Tensor Flow) and to compare and find out the model with the best accuracy rate. Here we compared the above three approaches based on Accuracy and time measurements using our own Image data. We have designed a game with multiple levels where in the object images are hidden in the background and the users have to find the image. Once the user selects the image the same image is sent to one of the three models to find out the model name and the result obtained is compared with the predefined values. For every correct selection the score increments and the corresponding name is lighten up. We have included 3 such levels and on successful completion of a single level user is moved on to next level. For the users convenience we have included an Interactive Question Answering Google conversation API that can be useful in case of providing hints to the user in case of any help. And for this we have used certain tools like api.ai, Heroku and Mongo DB to save the results. The goal is to analyze a particular image selected by the player using Image Analysis in the three different approaches and produce the corresponding output. This output is then compared with the remaining models to find out the best accuracy rate.**
We performed a rigorous training on the dataset using these 3 models and finally we observed the highest accuracy rate of 98% with the Tensor flow model.
Index Terms: **Clarifai API, Spark, Tensor Flow, API.AI(Google Conversation).**

1. INTRODUCTION

In the current years, there is a fast development in the area of Artificial Intelligence. There is a huge amount of research done in the field of Machine Learning and lot more new strategies have been proposed in the recent times with which we can accomplish more than before. The main goal of our application is to challenge the user to find out the hidden objects lying uncovered in a place. This also helps player to get out of the tensions and to provide relaxation for a certain time.

In this project we considered a simple room structure as our use case and have collected a data set of images that usually appear in every living room of an individual. Here we will display a set of objects to the player, once the player clicks on image that image will be predicted by using Image Classification program. These predictions are then checked with the predefined list and if it matches we will generate success message.

2. RELATED WORK

We studied deeply about the above mentioned three different approaches as a background work to come up with better results. We implemented image classification on the collected data to predict the image and we will find the accuracy of the selected images. Initially, we worked with the fundamentals of Spark, Tensor Flow after that we got to know the stream of data and usage.

Features:

- Image classification using the deep learning algorithms.
- Game to identify the hidden image.
- Score and Timer Clock are implemented for comparisons among players.

Data Source: Caltech 256 Data set,Canstockphoto.com Personally Captured images
Analytical Tools**:** Spark, Tensor Flow, Google Conversation API
Analytical Tasks: Predicting the images using image classification algorithms and showing the corresponding results.
Expected Input: Image selected by the user which gets processed and gives us the prediction of the image.
Output: Predictions are checked with the predefined list and if it matches we will generate success message.

3. PROPOSED WORK

Proposed Models: Our project guidelines include 3 models
Using Clarifai API – Clarifai API is a REST service which is used for image classification. Here in this case we don't perform any training on the dataset. We pass the user selected image into the Clarifai API and get back the image class dataset it belongs to.

Using Machine Learning – Random forest is a regression model that helps n image classification using the multiple regression to train a system.

Using Deep Learning –TensorFlow is an open source software library for machine learning across a range of tasks, and developed by Google to meet their needs for systems capable of building and training neural networks to detect and decipher

patterns and correlations, analogous to the learning and reasoning which humans use.

Algorithms and Pseudocode:

Image Classification Algorithm by Clarifai:

Clarifai API helps for video and image recognition as service.

This algorithm makes it simple for tags in a single step. This image classification helps in fast predict of images. This method is simple.

Random Forest Algorithm:

Random Forest is a classification and regression algorithm, originally designed for machine learning. It is an ensembled model, which uses the results from different models for calculations of response. It is easy to interpret the rules using the tree diagram of this model. Benefits of this algorithm: Accuracy, It runs effectively on the larger data sets.

```
val numClasses = 5
val categoricalFeaturesInfo = Map[Int, Int]()
val impurity = "gini"
val maxDepth = 5
val maxBins = 32
val featureSubsetStrategy = "auto"
val numTrees = 5

val model = RandomForest.trainClassifier(trainingData, numClasses,
categoricalFeaturesInfo, numTrees, featureSubsetStrategy, impurity,
maxDepth, maxBins)
```

Training the Classifier Based on the Random Forest:

```
println("numTrees " + numTrees + " featureSubsetStrategy " + featureSubsetStrategy +
  " impurity " + impurity + " maxDepth " + maxDepth)

val model = RandomForest.trainClassifier(training, numClasses, categoricalFeaturesInfo,
  numTrees, featureSubsetStrategy, impurity, maxDepth, maxBins)

val predictionAndLabel = test.map { point =>
  val prediction = model.predict(point.features)
  (point.label, prediction)
}

val testErr = predictionAndLabel.filter(r => r._1 != r._2).count.toDouble / test.count()
println("Test Error = " + testErr)
ModelEvaluation.evaluateModel(predictionAndLabel)
```

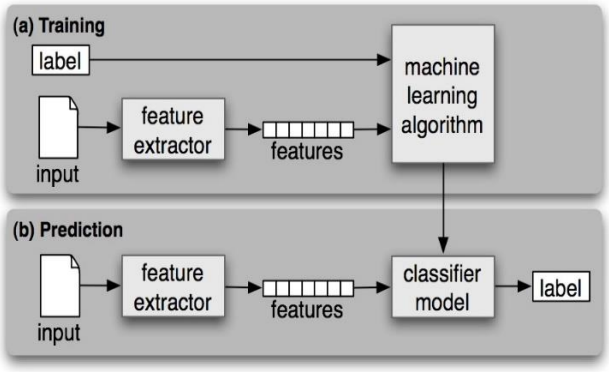
- Deep learning algorithm using inception v3:

Inception v3 is designed by google for image recognition, using the google inception architecture. Multiple iterations of trial and error results in high accuracy for the training of images using this inception model. This model use the multiple scoping mechanisms of tensor flow.

4. IMPLEMENTATION AND EVALUATION

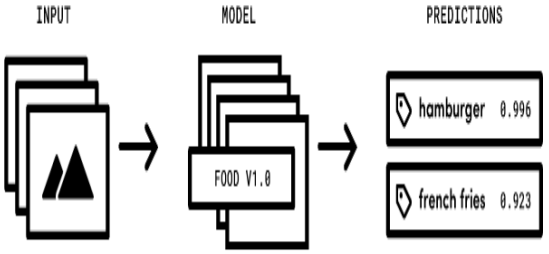
I. System Design and Implementation

1. Software Architecture

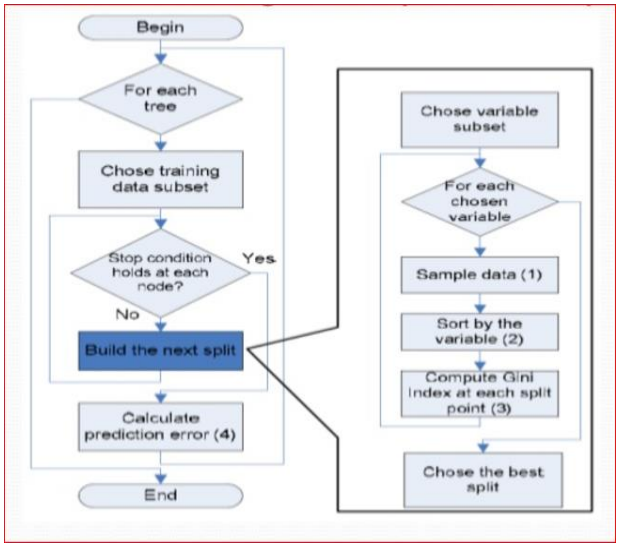


Architecture Flow:

Clarifai API:

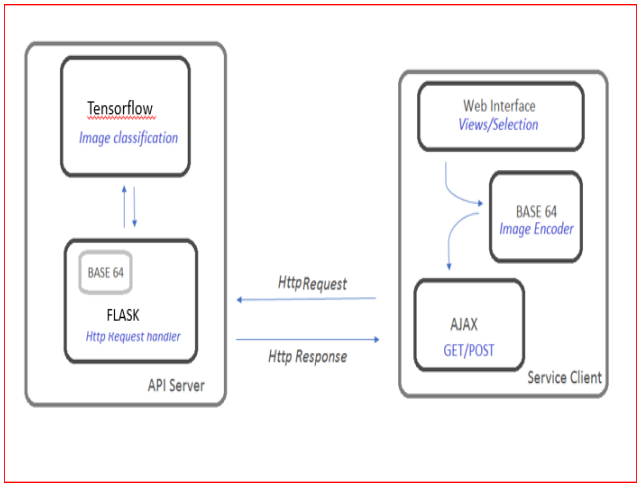


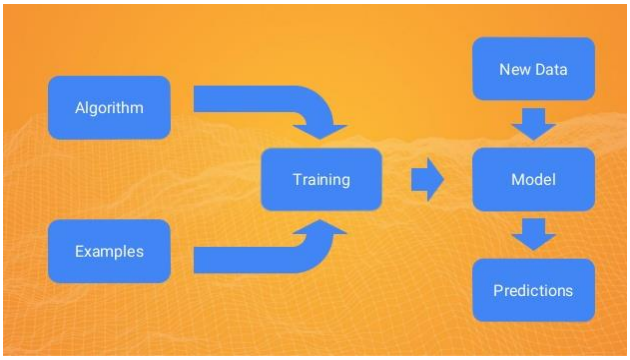
Machine learning – Random Forest Algorithm:



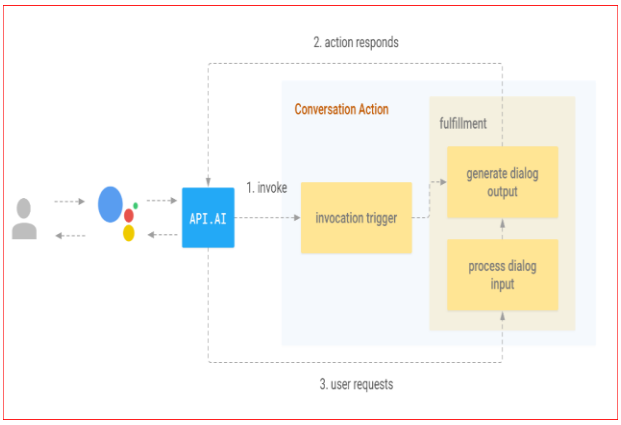
Deep Learning with Tensor Flow:

Architecture:



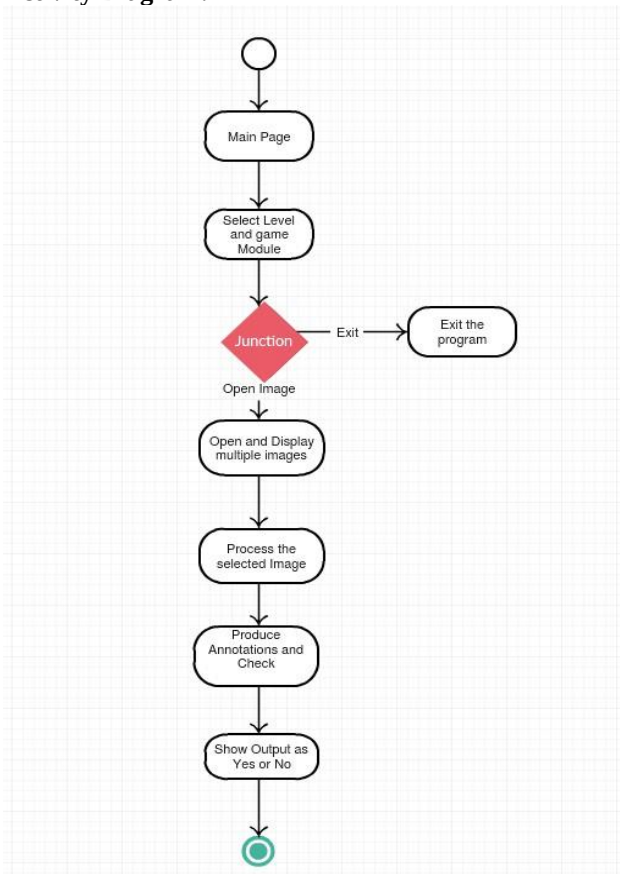


Google Conversation:



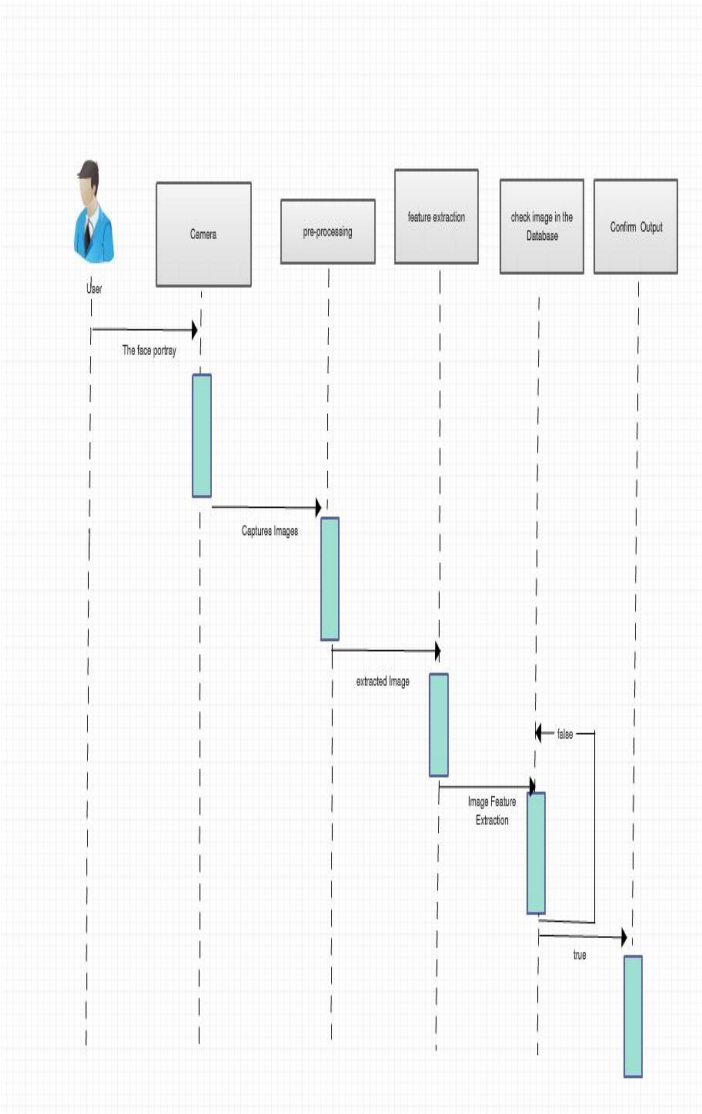
2. UML diagrams:

Activity Diagram:



In UML, Activity diagram is one of the important diagram, which defines dynamic features of application. The above is the activity diagram for our web application. Initially, we get the main page of the web application from there we can go to the level page of the game. In that level page we can select the game level or if we want to quit the game we can go to the quit option. By selecting the game level, image will be displayed with multiple unhidden images. User identify the listed images in that main displayed image and by clicking on the unhidden images, then selected images will be predicted by using Image classification techniques. As a result corresponding accuracy will be calculated in the three different techniques. Finally, if the user clicked image matches with the listed images, then a success message will be generated.

Sequence Diagram:



A sequence diagram is another UML interaction diagram, which demonstrates how things work with one another and in what order. The above is the sequence diagram for our web application, which shows object interactions set in time order.

Operation Specification

Input: User selected image

Output: Success message

We take the user selected image as input and we will apply some image classification techniques on that image in order to get the accuracy of that image. Image will be predicted and the results will be sent back to the web application to check with the list displayed in the web application. If that matches with the listed images then a success message will be displayed.

3. Implementation details:

As a part of the application we have implemented a web application which contains start page and level page. In this we have implemented the Image classification for the dataset in order to get the objects accuracy. In this web application, there would be some hidden images and player should identify the images based on the image names which are displayed on the game screen. This game has three different levels like Level 1, Level 2 and Level 3. This application has hint option on the game screen, by using this option player can get the clue about the image, which the player feels difficult to find the image in the screen. There is a timer running on the screen which can shows the application running time and if the timer reaches 5 minutes then automatically the game will be ended and the score will be displayed.

II. Evaluation and Results:

1. Evaluation plan

a. Datasets

Here we considered a set of images that usually exist in a living room of any individual.

We have captured images from different data sources like:

- Caltech 256 Data set
- Canstockphoto.com
- Personally Captured images

b. System Specifications:

- Providing the game with different levels.
- Displaying the hidden objects based on the game level complexity.
- On finding the hidden objects then predict the accuracy of the images.
- If it matches with the list which will provide on the screen, then the score will be increased.
- Once the given time is done, automatically the game will quit.

c. Measurements:

We measure the accuracy of predicting the image using three different methods External API (clarifai), Machine learning () and Tensor flow and suggest the best method of predicting the image based on the accuracy.

2. Evaluation & Results:

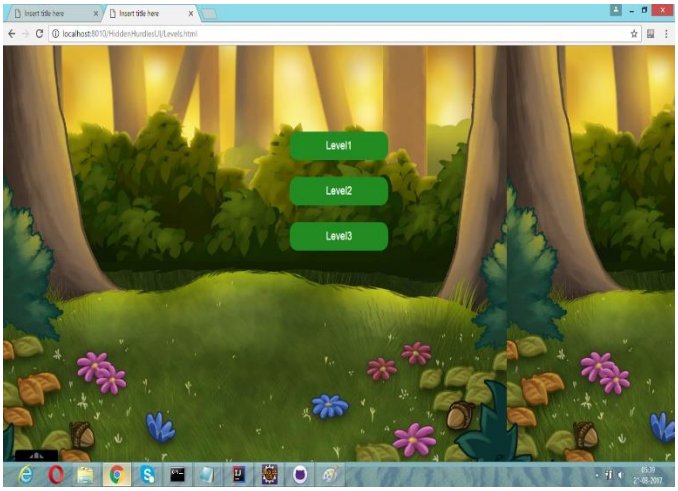
a. Comparative Evaluation:

I. Comparative evaluation for three different approaches:

Here we have done the image classification using the three different techniques like Clarifai, Machine learning and Tensor Flow to predict the image to get the predictions of the selected image and the accuracy of that particular image.

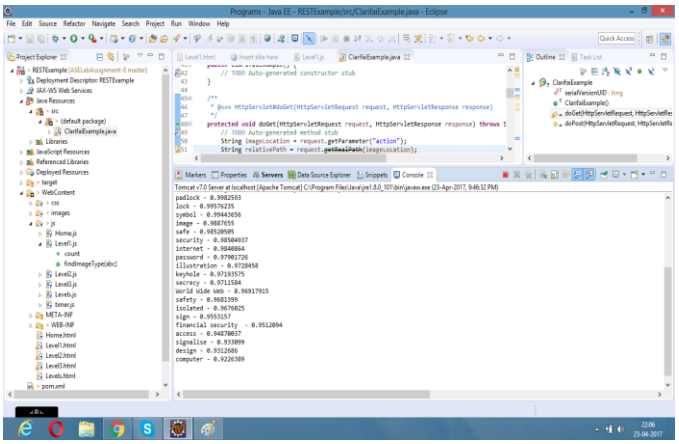
From the observations, we got to know that by using Tensor Flow Inception model, we can get the highest accuracy like 98% than the remaining techniques.

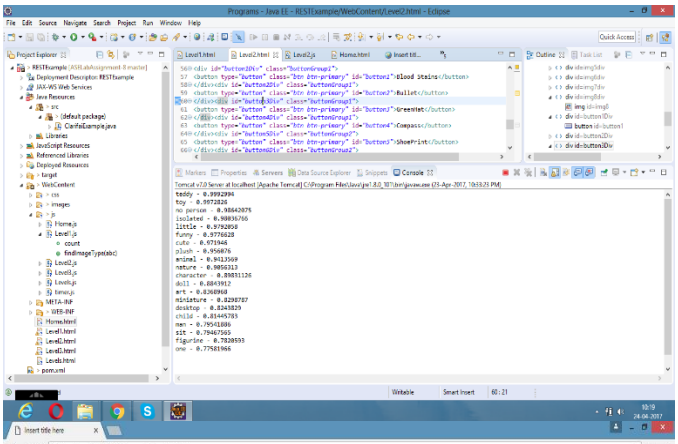
Web Application:

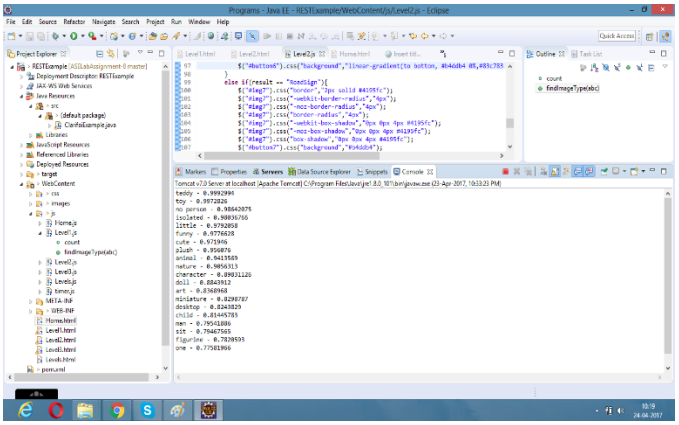


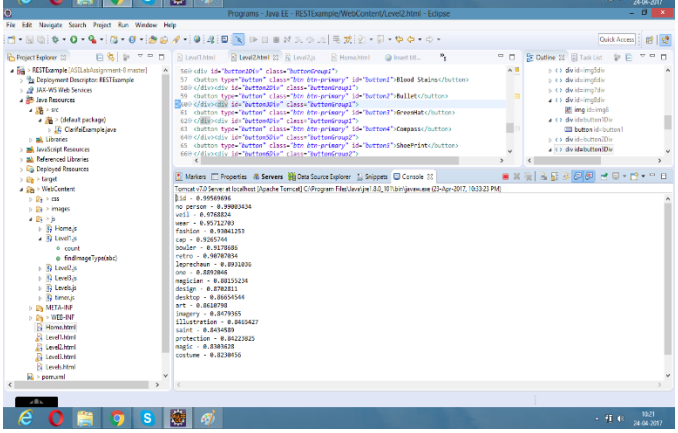
Clarifai API:

Level1:





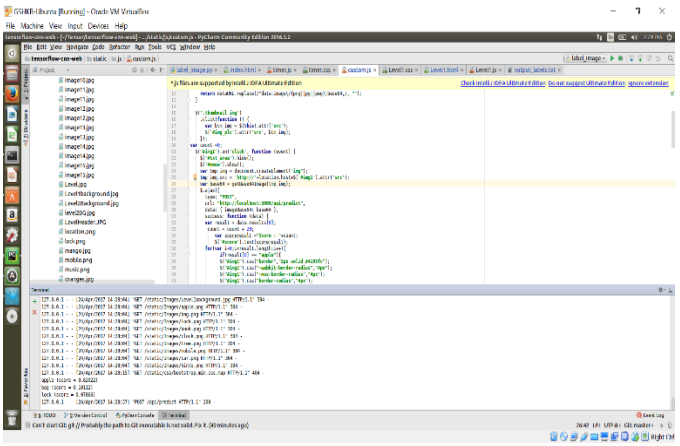




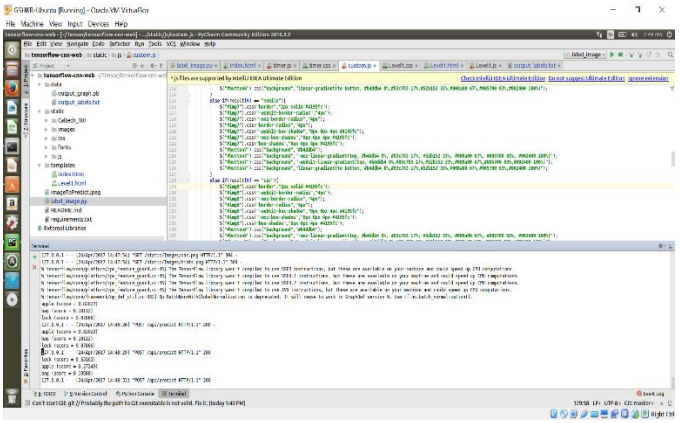
Machine Leaning (Random Forest):

Level2:

Level1:

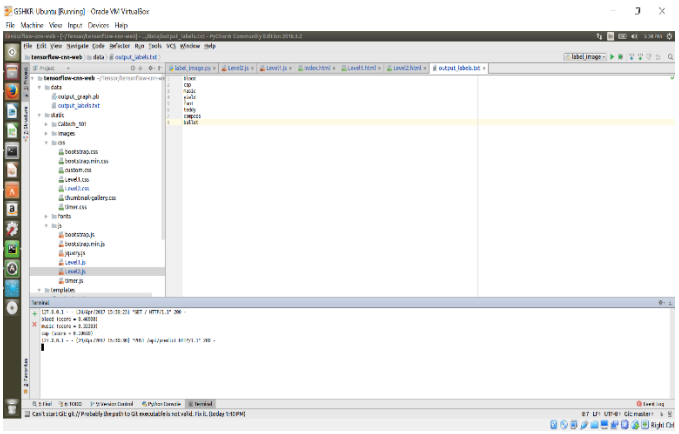


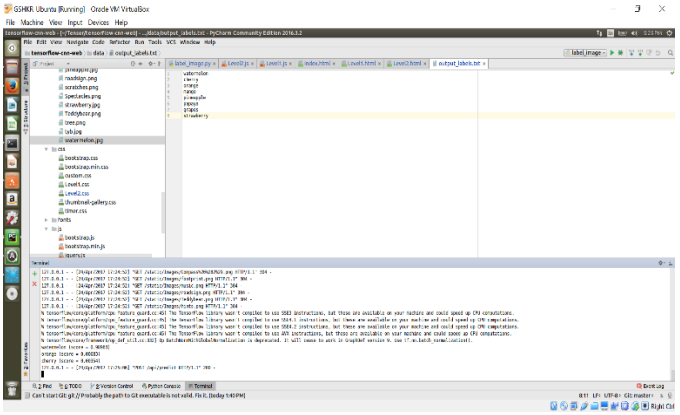
Level2:

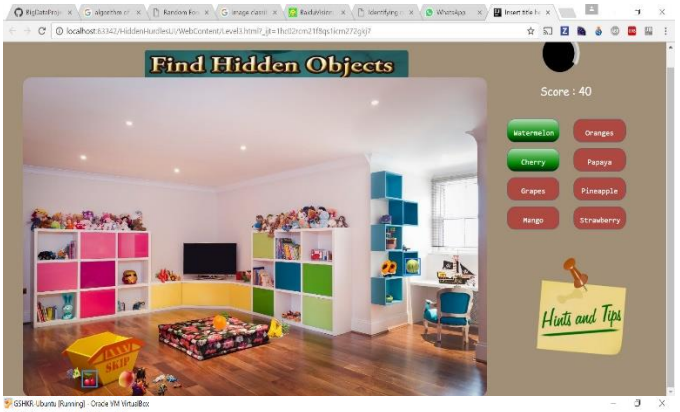


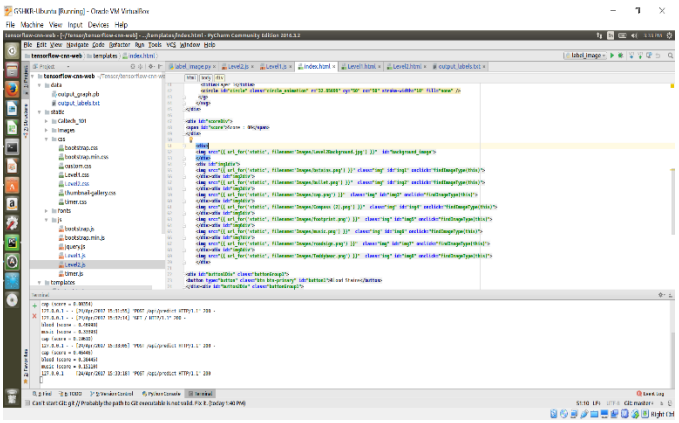
Tensor Flow (Inception Model):

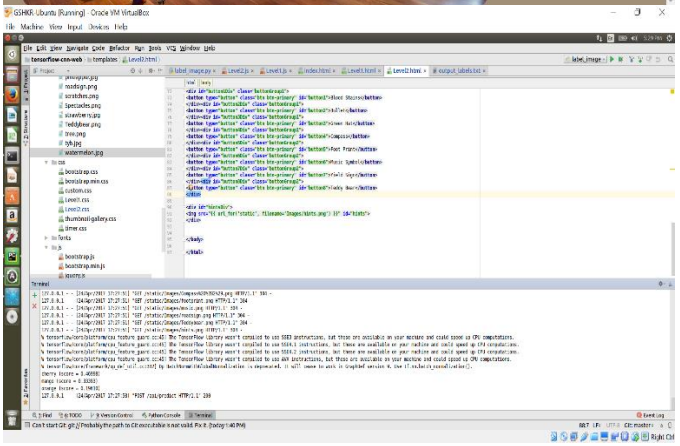
Level2:





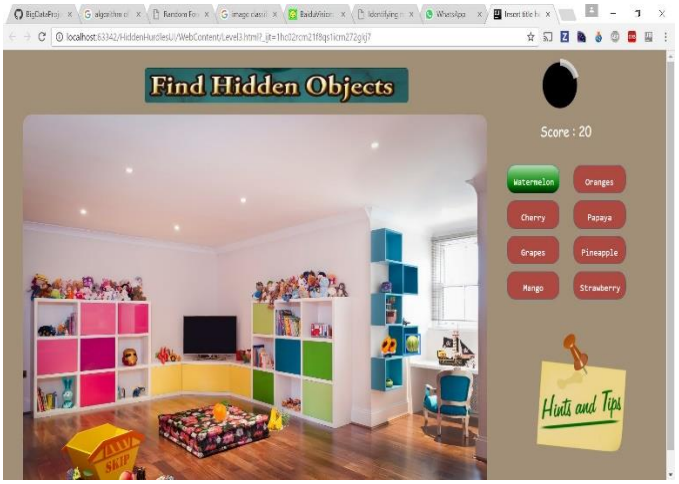






Level3:

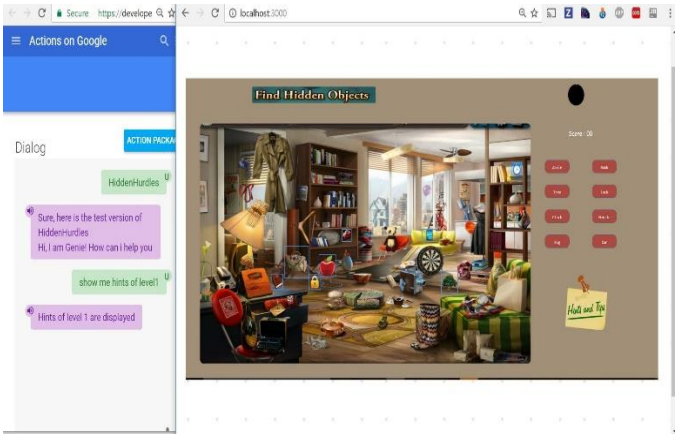
Hints:

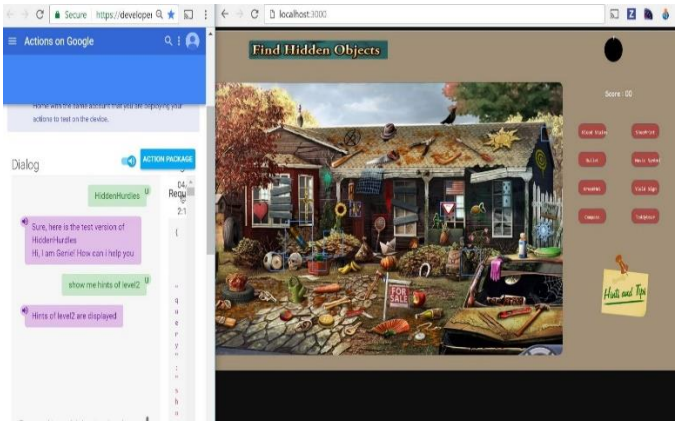


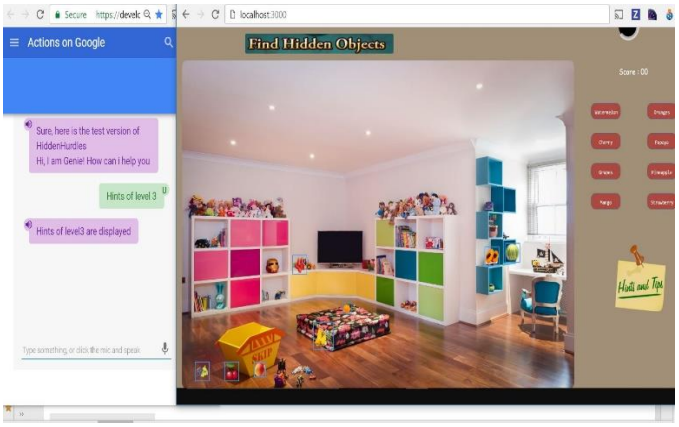
Level1:

Level2:

Google Conversation API:







II. Comparative evaluation with others:

We can predict the type of image with higher accuracy when compared to other approaches

4. DISCUSSION AND LIMITATIONS

limitations of the study:

Taking into consideration the resources we have i.e. the GPU and CPU we are narrowed to use only a small dataset which resulted in a lesser accuracy than expected. This can be enhanced further with the usage of powerful machines. Training the huge amount of data is time consuming. Training the data sets using Tensor Flow consumes lots of GPU memory which is expensive. We can't get the 100% of accuracy with the data sets.

5. CONCLUSION

We have performed the image classification using three models namely External API (Clarifai), Machine Learning Algorithm (Random Forest) and Tensor Flow (Inception Model) and predicted the type of image and obtained the accuracy of predicting the type of image correctly. We got an accuracy of 88% using Clarifai API on an average and for machine learning using random forest we got an accuracy of 92% on an average and finally using tensor flow inception model we got an accuracy of 98% on an average for all the images. By comparing the accuracy of the three methods we conclude that we get maximum accuracy for the using tensor flow Inception model hence we conclude tensor flow would predict an image with maximum accuracy when compared to Clarifai and Machine Learning.

MODEL	ACCURACY	Time Taken To Execute
Clarifai	88%	1.45sec.
Machine Learning	92%	1.69sec
Tensorflow	98%	2.10sec.

Table: Results for three different models.

REFERENCES

[1]. https://developer.clarifai.com/
[2]. https://spark.apache.org/docs/2.1.0/mllib-ensembles.html
[3]. https://www.tensorflow.org/tutorials/image_recognition
[4].https://research.googleblog.com/2016/03/train-your-own-image-classifier-with.html
[5].https://medium.com/@ageitgey/machine-learning-is-fun-part-3-deep-learning-and-convolutional-neural-networks-f40359318721
[6] https://en.wikipedia.org/wiki/Random_forest
[7] http://download.tensorflow.org/paper/whitepaper2015.pdf
[8] http://www.jmlr.org/papers/volume13/biau12a/biau12a.pdf
[9] https://research.google.com/pubs/pub45381.html