# Designing A Game for Locating Objects in Images

Project report submitted in partial fulfilment of the requirement for the degree of Bachelor of Technology

in

**Computer Science and Engineering/Information Technology**

By

Avichal Tyagi(121216)

Under the supervision of

Mr. Arvind Kumar

to


Department of Computer Science & Engineering

**Jaypee University of Information Technology Waknaghat, Solan-173234, Himachal Pradesh**

# Certificate

## Candidate's Declaration

I hereby declare that the work presented in this report entitled **"Designing A Game for Locating Objects in Images"** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering/Information Technology** submitted in the department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried out over a period from August 2015 to December 2015 under the supervision of **Mr. Arvind Kumar** .

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Avichal Tyagi 121216

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Mr. Arvind Kumar

_____

Dated:

# Acknowledgement

I gratefully acknowledge the Management and Administration of Jaypee University of Information Technology for providing me the opportunity and hence the environment to initiate and complete my project.

Date:

# Table of Content

# List of Figures

# Abstract

We introduce "Hunch Man", an entertaining web-based game that can help computers locate objects in images. People play the game because of its entertainment value, and as a side effect of them playing, we collect valuable image metadata, such as which pixels belong to which object in the image. The collected data could be applied towards constructing more accurate computer vision algorithms, which require massive amounts of training and testing data not currently available

# CHAPTER 1

# INTRODUCTION

## 1.1 Introduction :

Humans understand and analyze everyday images with little effort: what objects are in the image, where they are located, what is the background, what is the foreground, etc. Computers, on the other hand, still have trouble with such basic visual tasks as reading distorted text or finding where in the image a simple object is located. Although researchers have proposed and tested many impressive algorithms for computer vision, none have been made to work reliably and generally.

The only method currently available for obtaining precise image descriptions is manual labelling, which is tedious and thus extremely costly. But, what if people labelled images without realizing they were doing so? What if the experience was enjoyable? In this report we introduce a new interactive system in the form of a game with a unique property: the people who play the game label images for us. The labels generated by our game can be useful for a variety of applications. For accessibility purposes, visually impaired individuals surfing the Web need textual descriptions of images to be read aloud. For computer vision research, large databases of labelled images are needed as training sets for machine learning algorithms. For image search over the Web and inappropriate content filtering, proper labels could dramatically increase the accuracy of current systems.

Most of the best approaches for computer vision rely on machine learning: train an algorithm to perform a visual task by showing it example images in which the task has already been performed. For example, training an algorithm for testing whether an image contains a dog would involve presenting it with multiple images of dogs, each annotated with the precise location of the dog in the image. After processing enough images, the algorithm learns to find dogs in arbitrary images. A major problem with this approach, however, is the lack of training data, which, obviously, must be prepared by hand. Databases for training computer

vision algorithms currently have hundreds or at best a few thousand images — orders of magnitude less than what is required.

## 1.2 Introduction : Meta Data

Image metadata is text information pertaining to an image file that is embedded into the file or contained in a separate file that is associated with it.

Image metadata includes details relevant to the image itself as well as information about its production. Some metadata is generated automatically by the the device capturing the image. Additional metadata may be added manually and edited through dedicated software or general image editing software such as GIMP or Adobe Photoshop. Metadata can also be added directly on some digital cameras.

Image metadata can be very useful for cataloguing and contextualizing visual information. Many visual artists find the features useful in providing data about themselves and their images.

Image metadata can also help protect intellectual property. It is important to note, however, that copyright information is not adequate protection as it can easily be stripped away. Also, as with other types of content, metadata security can be cumbersome, requiring extra measures to secure image metadata and protect it from unauthorized access.

The three main categories of image metadata are:

**Technical metadata** is mostly automatically generated by the camera. It includes camera details and settings such as aperture, shutter speed, ISO number, focal depth, dots per inch (DPI). Other automatically generated metadata include the camera brand and model, the date and time when the image was created and the GPS location where it was created.

**Descriptive metadata** is mostly added manually through imaging software by the photographer or someone managing the image. It includes the name of the image creator, keywords related to the image, captions, titles and comments, among many other possibilities. Effective descriptive metadata is what makes images more easily searchable.

**Administrative metadata** is mostly added manually. It includes usage and licensing rights, restrictions on reuse, contact information for the owner of the image.

## Storing Metadata

- Metadata can be stored either internally, in the same file or structure as the data (this is also called embedded metadata), or externally, in a separate file or field from the described data.
- **Internal storage** means metadata always travel as part of the data they describe; thus, metadata are always available with the data, and can be manipulated locally.
- **External storage** allows collocating metadata for all the contents, for example in a database, for more efficient searching and management. Redundancy can be avoided by normalizing the metadata's organization.

Several standardized formats of metadata exist, including: Information Interchange Model (IPTC), Extensible Metadata Platform (XMP), EXchangable Image File (Exif), Dublin CoreMetadata Initiative (DCMI) and Picture Licensing Universal System (PLUS).

## 1.3 Introduction : The Open Mind Initiative

Our work is similar in spirit to the Open Mind Initiative , a worldwide effort to develop “intelligent” software. Open Mind collects information from regular Internet users (referred to as “netizens”) and feeds it to machine learning algorithms. Volunteers participate by answering questions and teaching concepts to computer programs. Our method is similar to Open Mind in that we plan to use regular people on the Internet to label images for us. However, we put greater emphasis on our method being fun because of the scale of the problem that we want to solve. We don’t expect volunteers to label all images on the Web for us: we expect all images to be labelled because people want to play our game.

## 1.4 Introduction : Computer Vision

**Computer vision** is a field that includes methods for acquiring, processing, analyzing, and understanding images and, in general, high-dimensional data from the real world in order to produce numerical or symbolic information, *e.g.*, in the forms of decisions. A theme in the development of this field has been to duplicate the abilities of human vision by electronically perceiving and understanding an image. Understanding in this context means the transformation of visual images (the input of retina) into descriptions of world that can interface with other thought processes and elicit appropriate action. This image understanding can be seen as the disentangling of symbolic information from image data using models constructed with the aid of geometry, physics, statistics, and learning theory. Computer vision has also been described as the enterprise of automating and integrating a wide range of processes and representations for vision perception.

## Recognition

The classical problem in computer vision, image processing, and machine vision is that of determining whether or not the image data contains some specific object, feature, or activity. Different varieties of the recognition problem are described in the literature:

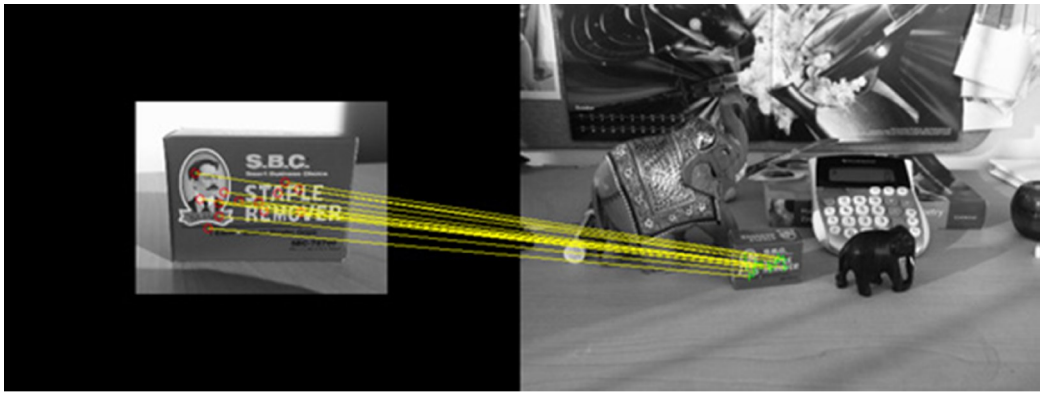
- **Object recognition** (also called **object classification**) – one or several pre-specified or learned objects or object classes can be recognized, usually together with their 2D positions in the image or 3D poses in the scene
- **Identification** – an individual instance of an object is recognized. Examples include identification of a specific person's face or fingerprint, identification of handwritten digits, or identification of a specific vehicle.
- **Detection** – the image data are scanned for a specific condition. Examples include detection of possible abnormal cells or tissues in medical images or detection of a vehicle in an automatic road toll system. Detection based on relatively simple and fast computations is sometimes used for finding smaller regions of interesting image data which can be further analyzed by more computationally demanding techniques to produce a correct interpretation.

## 1.5 Introduction : Object Detection in Computer Vision

Object detection is the process of finding instances of real-world objects such as faces, bicycles, and buildings in images or videos. Object detection algorithms typically use extracted features and learning algorithms to recognize instances of an object category. It is commonly used in applications such as image retrieval, security, surveillance, and automated vehicle parking systems.

You can detect objects using a variety of models, including:

Fig 1.1:Feature-based object detection

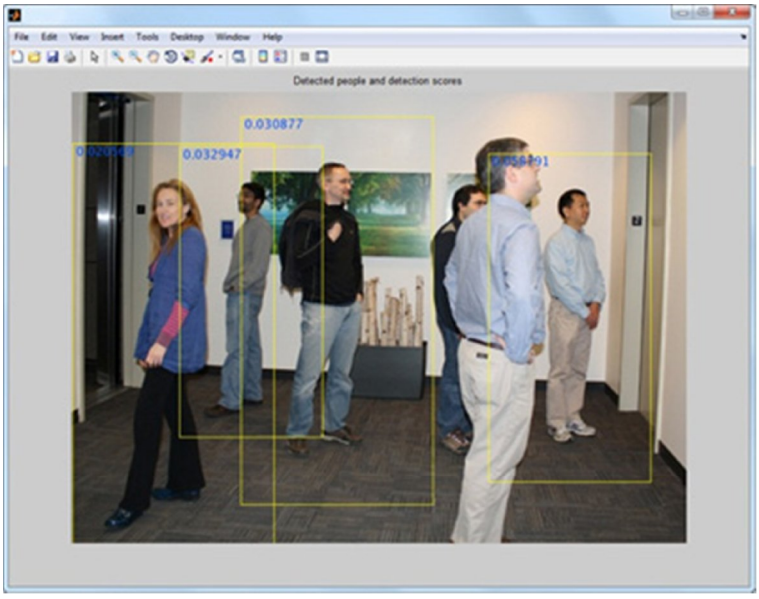



Detecting a reference object (left) in a cluttered scene (right) using feature extraction and matching. RANSAC is used to estimate the location of the object in the test image.

Fig 1.2:Viola-Jones object detection


Face detection (left) and stop sign detection (right) using the Viola-Jones Object Detector.

Fig 1.3:SVM classification with histograms of oriented gradients (HOG) features




Human detection using pretrained SVM with HOG features.

Fig 1.4:Image segmentation and blob analysis

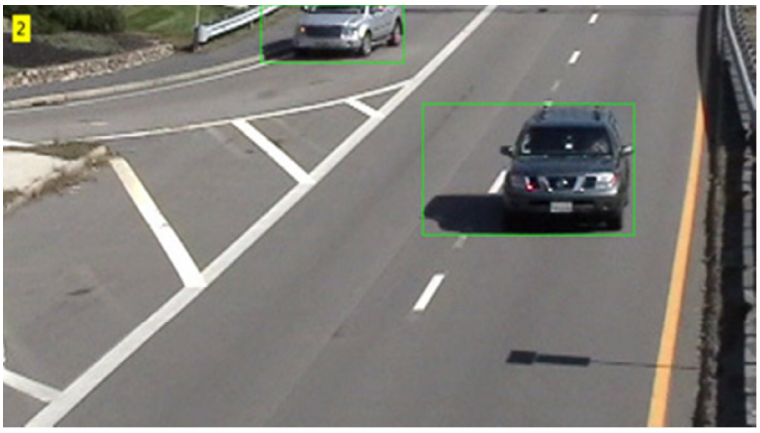


Fig 1.5:Moving cars are detected using blob analysis.

Image segmented using background subtraction. The moving pixels (foreground) detected from the video frame above are shown in white.

Other methods for detecting objects with computer vision include using gradient-based, derivative-based, and template matching approaches.

# CHAPTER 2

## Literature Review

## 2.1 Peekaboom: A Game for Locating Objects in Images

### BASIC GAME PLAY

[1]Peekaboom, as the name may suggest, is a game with two main components: “Peek” and “Boom.” Two random players from the Web participate by taking different roles in the game — when one player is Peek, the other is Boom. Peek starts out with a blank screen, while Boom starts with an image and a word related to it (see Figure 1). The goal of the game is for Boom to reveal parts of the image to Peek, so that Peek can guess the associated word. Boom reveals circular areas of the image by clicking. A click reveals an area with a 20-pixel radius. Peek, on the other hand, can enter guesses of what Boom’s word is. Boom can see Peek’s guesses and can indicate whether they are hot or cold. When Peek correctly guesses the word, the players get points and switch roles; play then proceeds on a new image-word pair. If the image-word pair is too difficult, the two players can “pass,” or opt out, of the current image. Passing creates the same effect as a correct guess from Peek, except that the players get no points. To maximize points, Boom has an incentive to reveal only the areas of the image necessary for Peek to guess the correct word. For example, if the image contains a car and a dog and the word associated to the image is “dog,” then Boom will reveal only those parts of the image that contain the dog. Thus, given an image-word pair, data from the game yield the area of the image pertaining to the word.

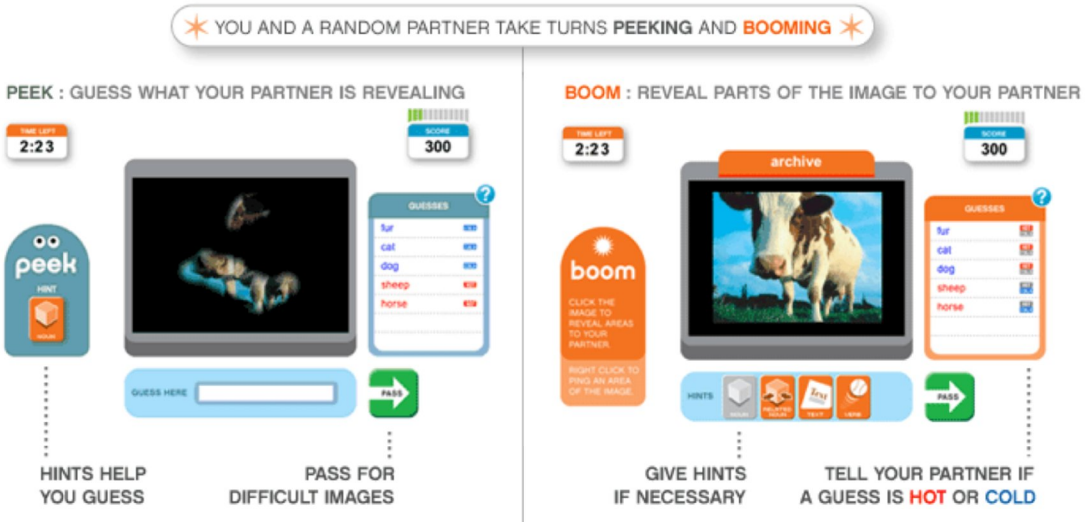



**Fig 2.1. Peek and Boom. Boom gets an image along with a word related to it, and must reveal parts of the image for Peek to guess the correct word. Peek can enter multiple guesses that Boom can see.**

The goal is to construct a database for training computer vision algorithms. Here we discuss exactly what information is collected by Peekaboom and how it is collected. On input an image-word pair , Peekaboom collects the following information:

• **How the word relates to the image.** Is it an object, person, or animal in the image, is it text in the image, is it a verb describing an action in the image, is it an object, person, or animal not in the image but related to it? The ESP Game associates words to images, but does not say how the word is related to the image. Hint buttons in Peekaboom allow us to determine the relation of the word to the image. This is useful in multiple ways, but for the purposes of constructing training sets for computer vision, it allows us to weed out “related nouns” and to treat “text” separately.

• **Pixels necessary to guess the word.** When Peek enters the correct word, the area that Boom has revealed is precisely enough to guess the word. That is, we can learn exactly what context is necessary to determine what the word refers to. This context information is absolutely necessary when attempting to determine what type of object a set of pixels constitutes

• **The pixels inside the object, animal, or person.** If the word is a noun directly referring to something in the image, “pings” give us pixels that are inside the object, person, or animal.

• **The most salient aspects of the objects in the image.** By inspecting the sequence of Boom’s clicks, we gain information about what parts of the image are salient with respect to the word. Boom typically reveals the most salient parts of the image first (e.g., face of a dog instead of the legs, etc.).

• **Elimination of poor image-word pairs.** If many independent pairs of players agree to pass on an image without taking action on it, then likely they found it impossibly hard because of poor picture quality or a dubious relation between the image and its label. By implementing an eviction policy for images that we discover are “bad,” we can improve the quality of the data collected (as well as the fun level of the game). When multiple players have gone through the same image, these pieces of information can be combined intelligently to give extremely accurate and useful annotations for computer vision. Later in the paper, for example, we show how a simple algorithm can use the data produced by Peekaboom to calculate accurate object bounding-boxes

**Figure 2.2. The image on the left contains a car driving through the street, while the one on the right has a person crossing the same street. Both the car and the person are exactly the same set of pixels up to a rotation by 90 degrees**

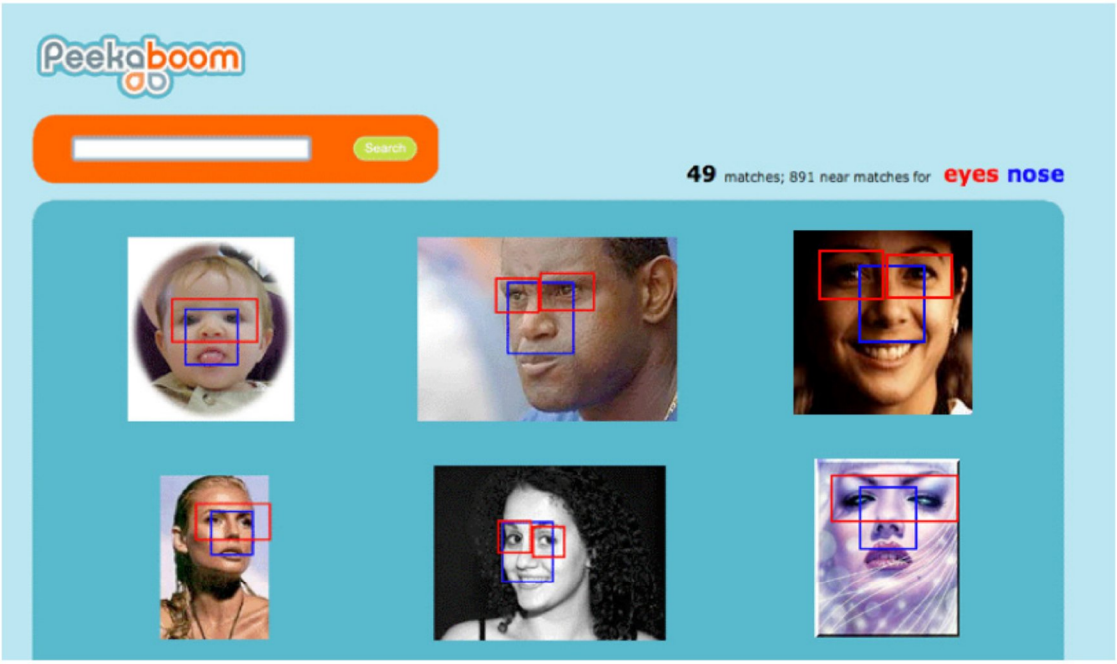



**Figure 2.3. Object bounding-boxes obtained from Peekaboom data.**

## 2.2 The ESP Game: Labelling Images with a Computer game

[2]The game is played by two partners and is meant to be played online by a large number of pairs at once. Partners are randomly assigned from among all the people playing the game. Players are not told whom their partners are, nor are they allowed to communicate with their partners. The only thing partners have in common is an image they can both see. From the player's perspective, the goal of the ESP game is to guess what their partner is typing for each image. Once both players have typed the same string, they move on to the next image (both player's don't have to type the string *at the same time*, but each must type the same string at some point while the image is on the screen). We call the process of typing the same string "agreeing on an image".


**Fig 2.2.1.Partners agreeing on an image. Neither of them can see the other's guess**

Partners strive to agree on as many images as they can in 2.5 minutes. Every time two partners agree on an image, they get a certain number of points. If they agree on 15 images they get a large number of bonus points. The thermometer at the bottom of the screen (see Figure 2) indicates the number of images that the partners have agreed on. By providing players with points for each image and bonus points for completing a set of images, we reinforce their incremental success in the game and thus encourage them to continue playing. Players can also choose to pass or opt out on difficult images. If a player clicks the pass button, a message is generated on their partner's screen; a pair cannot pass on an image until both have hit the pass button. Since the players can't communicate and don't know anything about each other, the easiest way for both players to type the same string is by typing something related to the common image. Notice, however, that the game doesn't ask the

players to describe the image: all they are told is that they have to “think like each other” and type the same string (thus the name “ESP”).

## When is an Image “Done”?

As a particular image passes through the ESP game multiple times, it will accumulate several labels that people have agreed upon. The question is, at what point is an image considered to have been completely labelled and thus no longer used in the game? Our answer to this question is to remove an image from the game when it is no longer enjoyable to guess its contents with a partner. This will  list of taboo words, such that pairs are unable to agree on new labels and consistently ask their partners to pass on the image. Repeated passing notifies the system that an image should no longer be used for the game at that point in time. (Repeated passing might also indicate that the image is too complex to be used in the game, in which case the image should also be removed.) Fully labelled images are re-inserted into the game when several months have passed because the meaning of the images may have changed due to maturation effects. The English language changes over time, as do other languages .We want to capture the labels appropriate to an image, and thus if the language referring to that image changes over time, so should our labels. In addition to changes in language, cultural changes may occur since a particular image has last been labelled. Thus a picture of something or someone that was labelled as “cool” or “great” six months prior may no longer be considered to be so. For example, an image of Michael Jackson twenty years ago might have been labelled as “superstar” whereas today it might be labelled as “criminal.”.

# CHAPTER 3

## System Development

### 3.1 Game Basics

It is a web-based multiplayer game. Two players, a guesser and a picker, will be playing this game simultaneously. Picker will pick an object from the image. Guesser's task will be to identify the object chosen by the picker. Being a game, it will have some rules.

**Game Rules**

1. Only two players can play at once. Each player will get 5 turns to guess. So, there will be a total of 10 matches per game.
2. Picker will pick an object and send one hint to guesser. Hints cannot contain the word or words which are in object's name.
3. Hints cannot exceed 120 characters.
4. Hints cannot contain common chat words, such as "Hi", "Hello", "How are you?", etc.
5. Hints cannot contain taboo words.
6. Hints cannot contain special symbols.
7. Picker will be given 30 seconds to pick an object.
8. Guesser will be given 60 seconds to identify the object. The time limit can be exceeded by using game coins.
9. The points distribution will be as follows:
    a. If guesser is right at first guess, it gains 50 points.
    b. For each next try, guesser loses 5 points.
    c. There are no negative points or gold coins. Minimum limit is zero.
    d. If guesser asks for another hint, it loses 10 points. If guesser don't want to lose points, it can use game coins.
    e. Picker gains 10 points if guesser is right at first try.
10. Guesser can ask for more hints from picker, but it will lose points. Guesser can only ask for maximum of 3 hints per match.
11. In case guesser thinks that the object chosen by picker is not in the picture, guesser can challenge the picker. If that object is tagged to that picture, challenge will fail and guesser will lose points. If that object is not tagged to that picture, the challenge will

be reported to administrator. If administrator thinks that object is in that picture, the challenge will fail and guesser will be notified. In this case, nothing will happen. If guesser's challenge is found right by administrator, guesser will gain some gold coins and picker will lose some gold coins as well as get warning.

12. Each player will get warnings if they are reported. If warnings exceed five times, the player will be banned permanently from the game.

## 3.2 Game Implementation:

**Tools**:

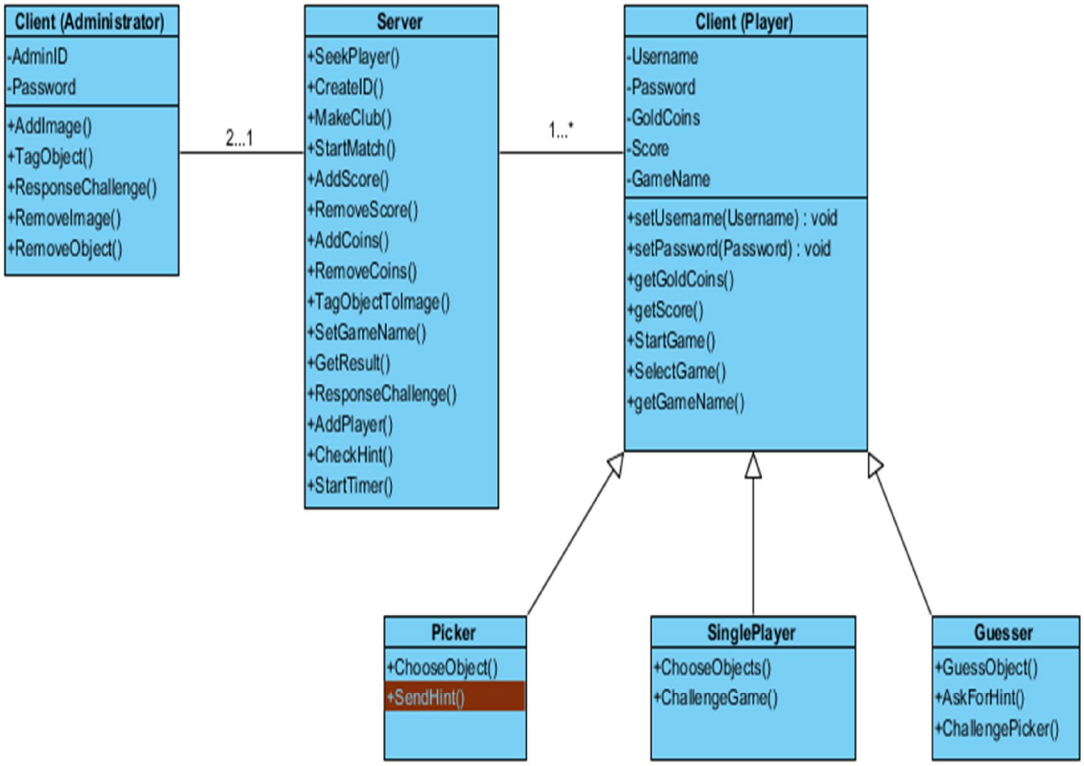
We will be using Visual Studio 2015 for this project. This web-based game will be developed using Visual Basics C# express . Following are the reasons for preferring :

1. .NET framework is very organized and systematic. Unlike java, it's very easy to maintain all the programs in their respective locations in .NET framework.
2. Visual Studio is very advanced IDE than other IDE's like Eclipse, NetBeans, etc. All the files of project are well organized already. We have no need to worry about their locations. Visual Studio takes care of such things by itself.
3. .NET library is very vast. Unlike Java, we have no need to download and add additional packages and libraries.
4. ASP.NET MVC is great for creating web applications. MVC stands for Models, Views and Controllers.

- Models: Models are basically classes which are used to describe the properties in a web page. In other words, a web page contains lots of elements for user interaction, for example, a web form asking user its user name. The user name is an element in web page whose value will be entered by the user and stored in a variable. This variable will be declared in model to make it easy to use in other programs in the project.
- View: View is basically the HTML or HTML5 page which user sees on the browser. It is automatically generated by MVC according to the model.
- Controller: Controller acts as a bridge between models and respective views.

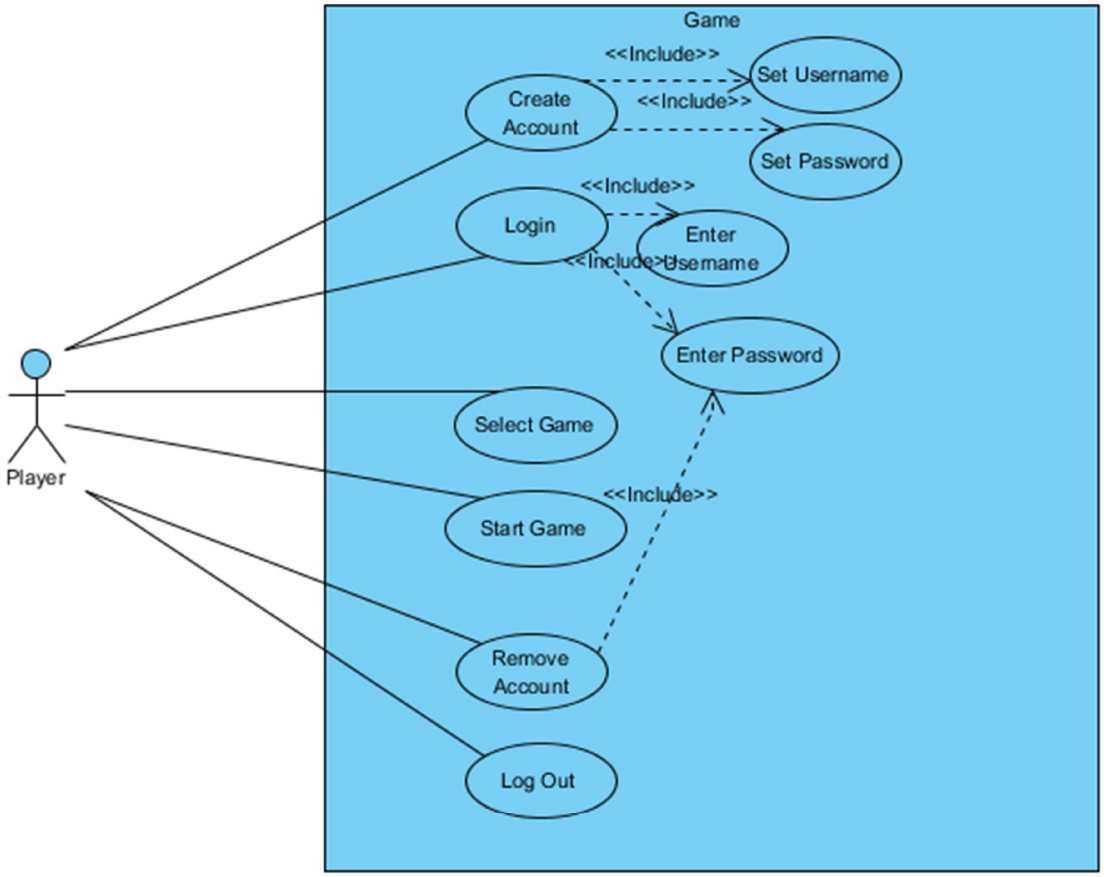
5. Since Visual Studio creates most of the code for us automatically, we only have to concentrate on the algorithms we are using in our project and its design. Unlike Java, we have no need to worry about mapping between classes and web pages, organization structure and libraries.

## 3.3 Class Diagrams:

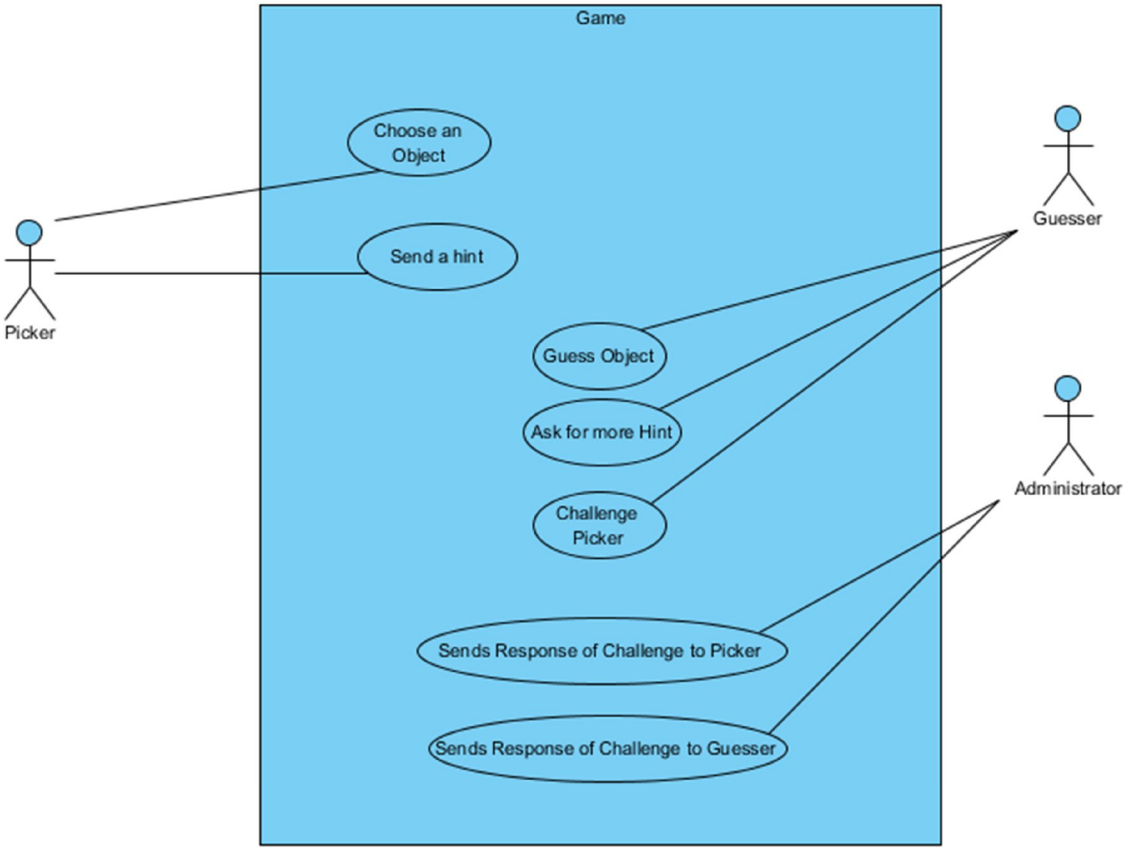


## 3.4 Use Case Diagrams:

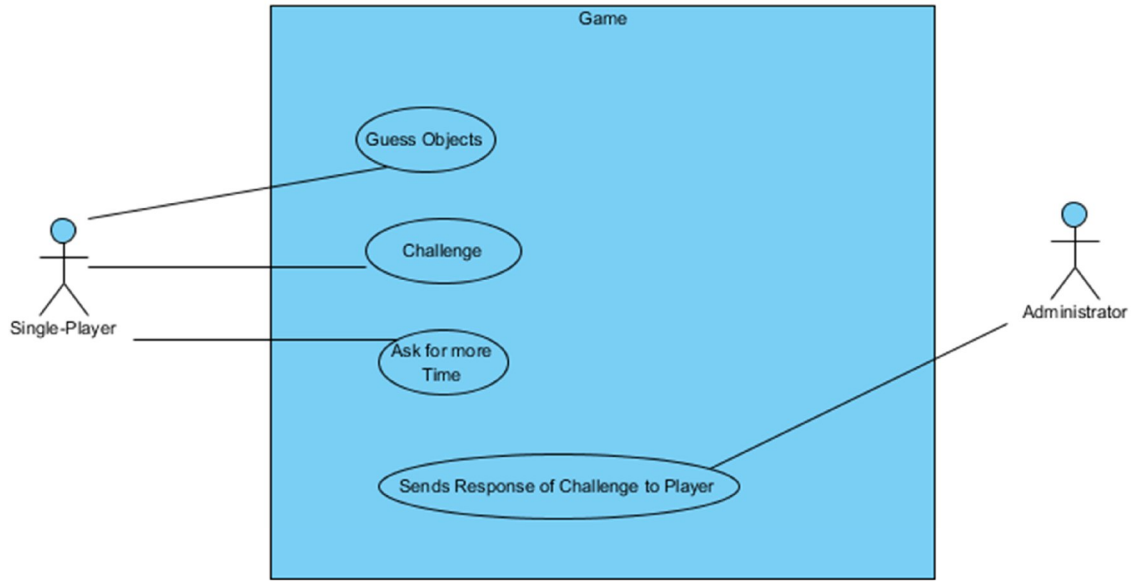
**Player**:



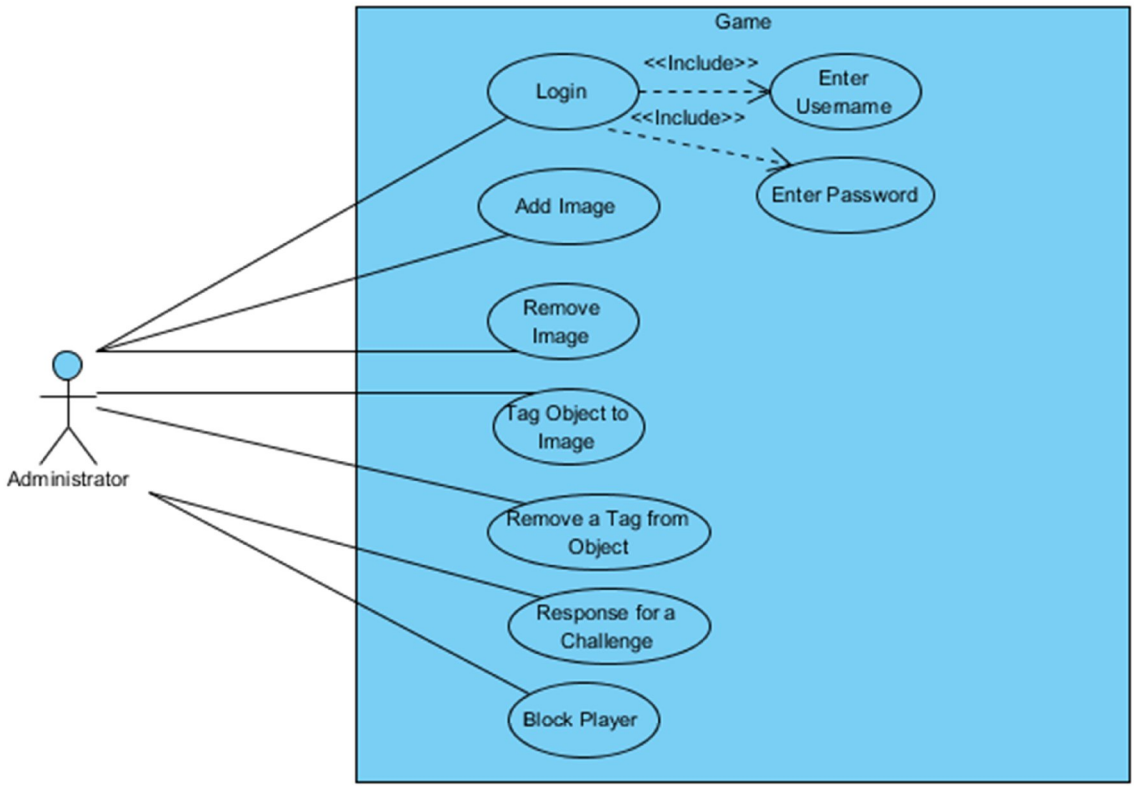
**Picker and Guesser:**



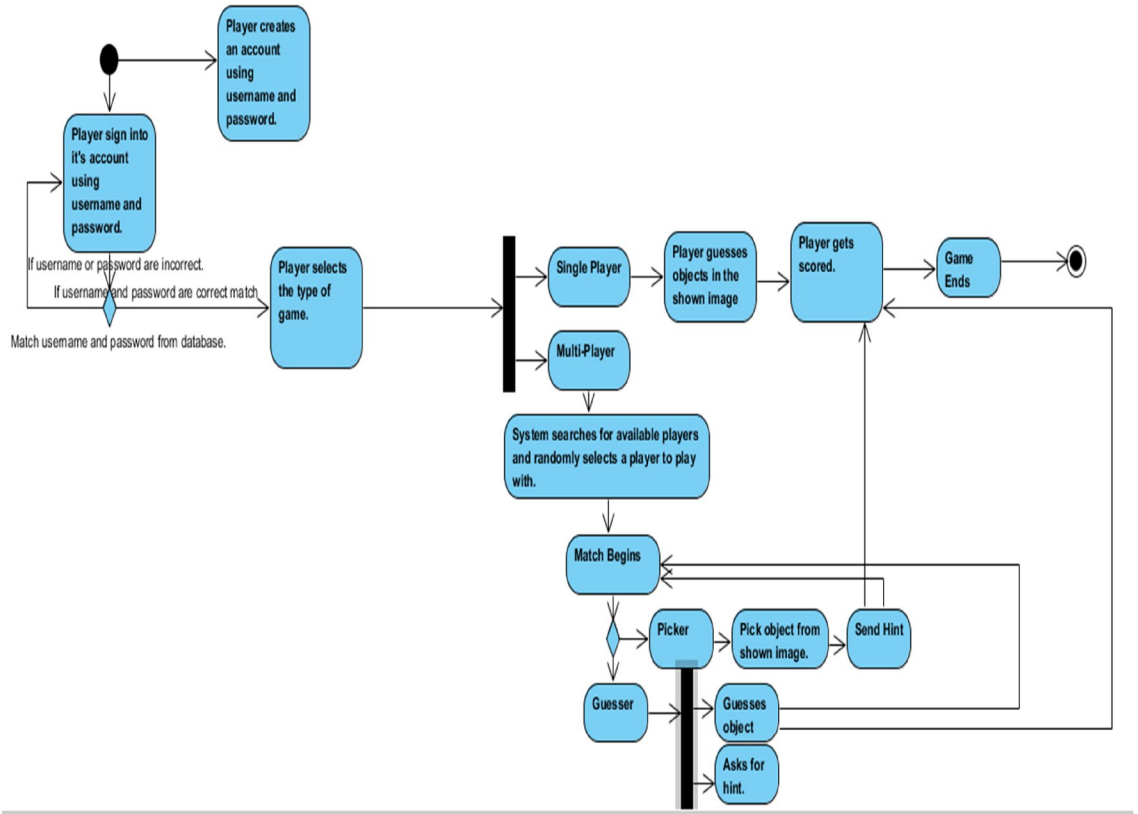

**Single Player**:



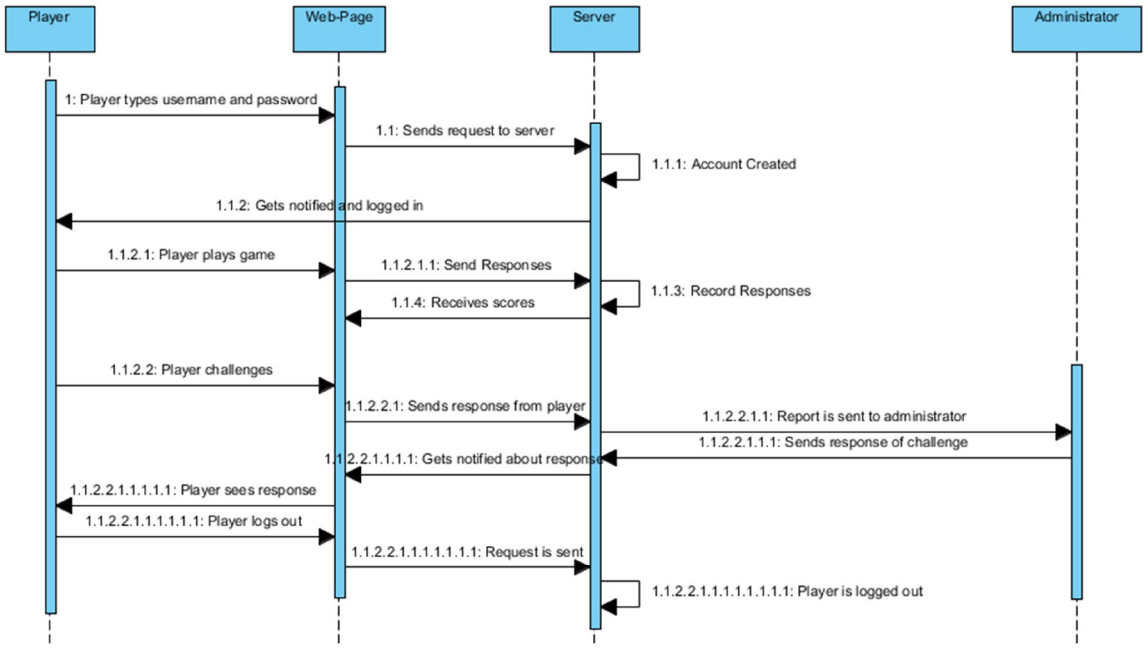
**Administrator**:



## Activity Diagram:




## Sequence Diagram:



## 3.5 Algorithms and Implementation:

1. A player must have an account to play game. Unique username and password will be required to create new account. Each player's username and password will be stored in a database.
2. This database of player will include following entities:
    a. Username
    b. Password
    c. One Time ID
    d. Scores
    e. Gold Coins
    f. Warnings
3. Pictures will be stored in a separate database. Each picture will have tags. These tags are the objects present in the image.
4. The home page of application will show the option to login and create new account.
5. Login are of two types, player and administrator.
6. Account created will be only a player account.
7. When player logs in, it will be asked to select game.
8. There are two kinds of games, single-player and multi-player.
9. In single-player game, player will be asked to guess some objects in image shown. The objects guessed by user will be matched with tagged objects of that image. The player will earn 10 points for each correct objects and 0 points for incorrect objects. However, since it's a new database, player can challenge. The challenge will be observed by administrator and response will be sent accordingly. If object is found in image, the object will be tagged to image by administrator and 5 gold coins will be awarded to player. Otherwise, player will receive no rewards.
10. In multi-player game, following methods will be used:
    a. In Peek-a-Boom game, the players with nearby ip address could not play with each other because they could cheat together. Such idea is good, but it also limits the amount of players playing game together. A better option, which we think, could be an anonymous play. Players will not know with whom they are playing with. Such will be achieved by a game–name. A game-name will be assigned to the players. This name will be shown in the game for the players, instead of username. This name will be assigned according to player's game

activities and interests. The game-name will be very common, so, it is very unlikely for two persons to know with whom they are playing with.

b. When the player will start a multiplayer game, temporarily, a one-time id will be generated and assigned to the player. This id will depend on the total number of players registered. For example, if total number of players are 30, id will be of two digits. In case of 300, id will be of three digits.All these ids will be mapped to username using hash-maps. After every 2 seconds, a thread running in server will generate two random numbers and those players will be matched together for matches.
c. While playing game, a thread will be used for time. The time monitoring for each match will be observed by this thread.
d. Picker will think about the object and type its name in the text box provided. It can also choose from 3 options provided by server. It will type a hint in the provided text box for hint. Then, it will click on confirm. The object will be confirmed in the current thread in the server.
e. Guesser will get the hint using real-time messaging. According to hint, guesser will type response in the text box provided or click on button to ask for more hints. Internal time will not stop if guesser asks for more hints. When guesser confirms, guesser's response will be matched with picker's response. If both responses are a match, the object will be tagged to the image.
f. If guesser challenges picker, guesser will type the reason and confirm it. The reason will be recorded against picker.
g. The hints which picker sends to guesser will be recorded for whole game and if no challenge is observed, it will be deleted when thread gets released. However, if challenge is observed, the conversation will be recorded and sent to administrator for further actions.
h. In text boxes, regular expressions will be used to block chat words, for example, "Hi!", "How are you?", etc. Same will be used for blocking taboo words, symbols and numbers.
i. Administrator can add, view and delete images. It can also add tags to images or remove tags from images.
j. Administrator can block any player, if it finds the player to be notorious.

k. If the player is blocked, all the data information, like scores, gold coins, will be removed permanently. However, the player can create new account with same username again.

11. A player can remove its account on its own free will. If a player forgets password, it will be asked to change password.

# CHAPTER 4

## Performance Analysis:

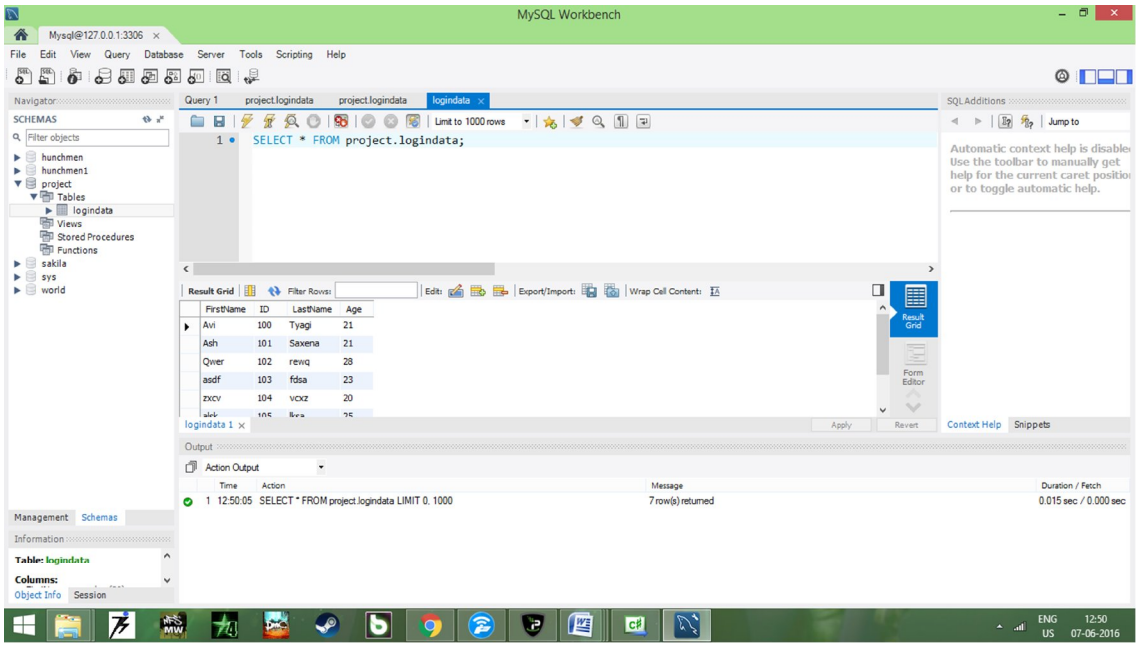



Fig4.1

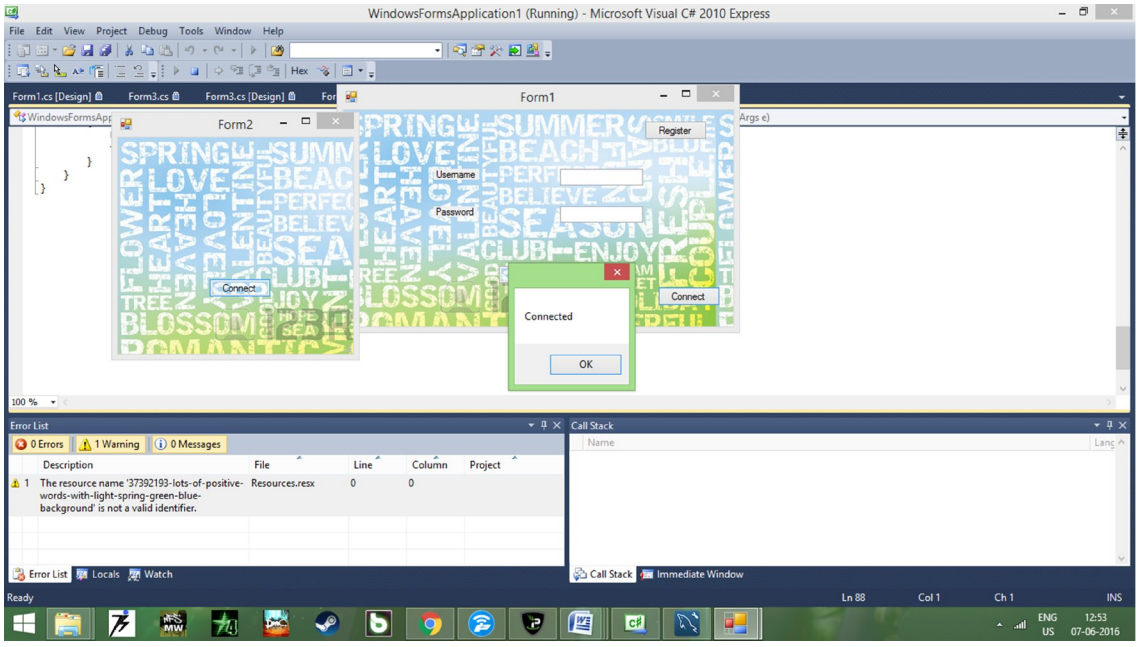



Fig4.2

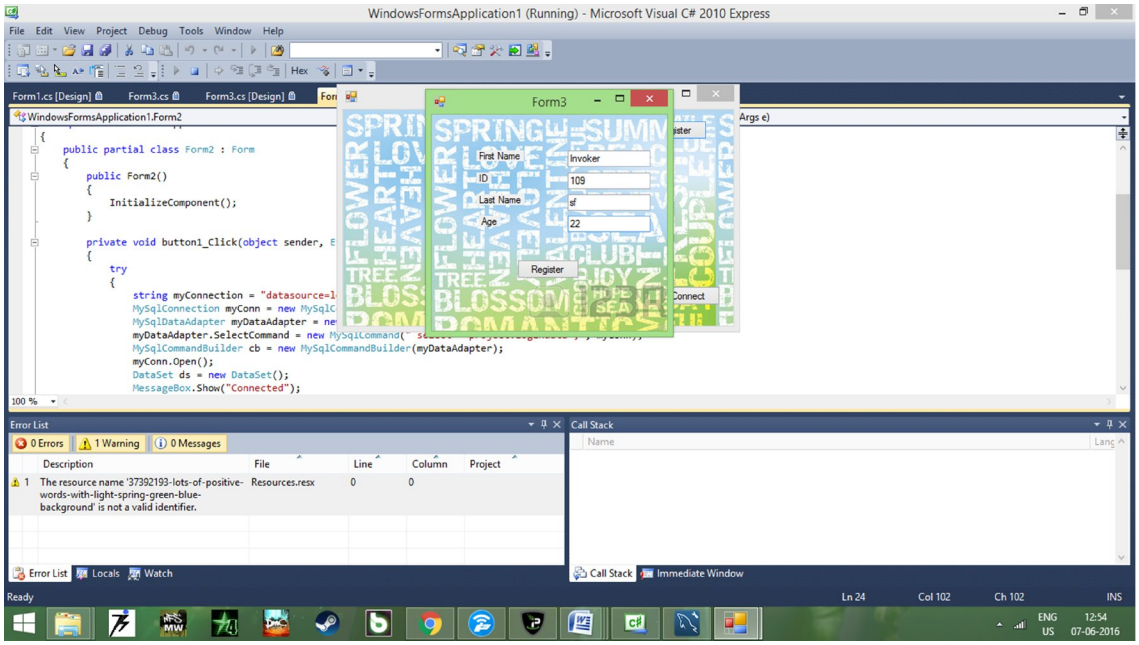

WindowsFormsApplication1 (Running) - Microsoft Visual C# 2010 Express
File Edit View Project Debug Tools Window Help
Form3
First Name
Invoker
ID
109
Last Name
sf
Age
22
Register
Connect
public partial class Form2 : Form
public Form2()
InitializeComponent();
private void button1_Click(object sender,
try
string myConnection = "datasource=1
MySqlConnection myConn = new MySqlC
MySqlDataAdapter myDataAdapter = ne
myDataAdapter.SelectCommand = new MySqlCommand(
MySqlCommandBuilder cb = new MySqlCommandBuilder(myDataAdapter);
myConn.Open();
DataSet ds = new DataSet();
MessageBox.Show("Connected");
Error List
0 Errors
1 Warning
0 Messages
Description
File
Line
Column
Project
The resource name '37392193-lots-of-positive-words-with-light-spring-green-blue-background' is not a valid identifier.
Resources.resx
Call Stack
Name
Error List
Locals
Watch
Call Stack
Immediate Window
Ready
Ln 24
Col 102
Ch 102
INS
ENG
US
12:54
07-06-2016

Fig 4.3

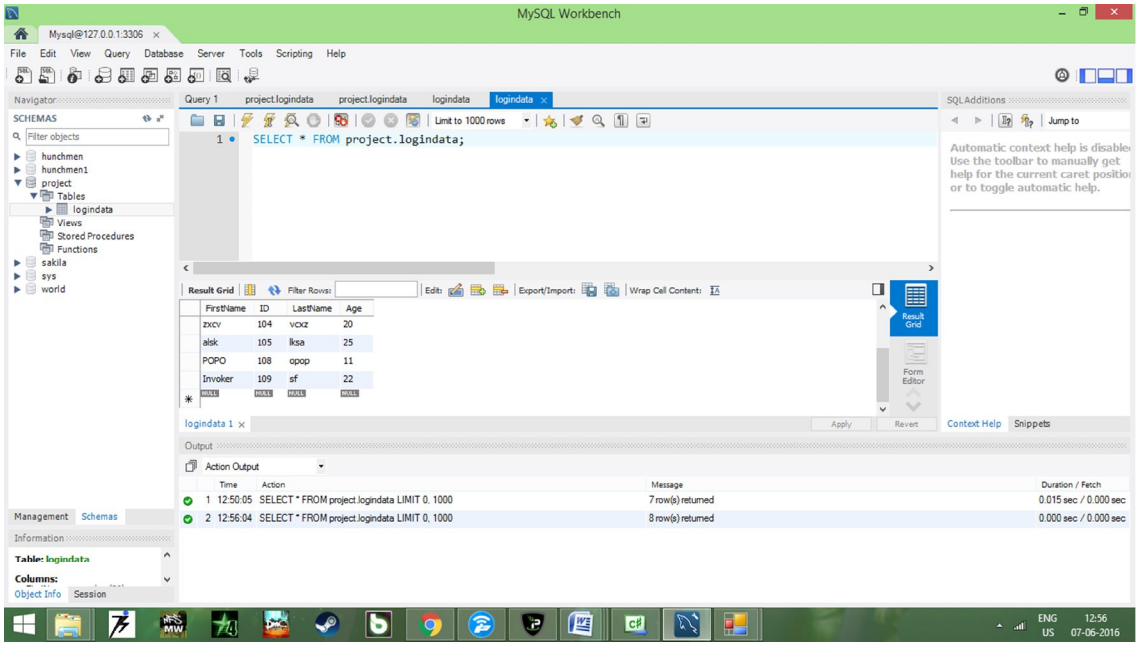

MySQL Workbench
Mysql@127.0.0.1:3306
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
hunchmen
hunchmen1
project
Tables
logindata
Views
Stored Procedures
Functions
sakila
sys
world
Query 1
project.logindata
project.logindata
logindata
logindata
Limit to 1000 rows
1 SELECT * FROM project.logindata;
SQL Additions
Jump to
Automatic context help is disable
Use the toolbar to manually get
help for the current caret positio
or to toggle automatic help.
Result Grid
Filter Rows:
Edit:
Export/Import:
Wrap Cell Content:
FirstName ID LastName Age
zxcv 104 vcxz 20
alsk 105 lksa 25
POPO 108 opop 11
Invoker 109 sf 22
Result Grid
Form Editor
logindata 1
Apply
Revert
Context Help
Snippets
Output
Action Output
Time Action Message Duration / Fetch
1 12:50:05 SELECT * FROM project.logindata LIMIT 0, 1000 7 row(s) returned 0.015 sec / 0.000 sec
2 12:56:04 SELECT * FROM project.logindata LIMIT 0, 1000 8 row(s) returned 0.000 sec / 0.000 sec
Management
Schemas
Information
Table: logindata
Columns:
Object Info
Session
ENG
US
12:56
07-06-2016

Fig 4.4

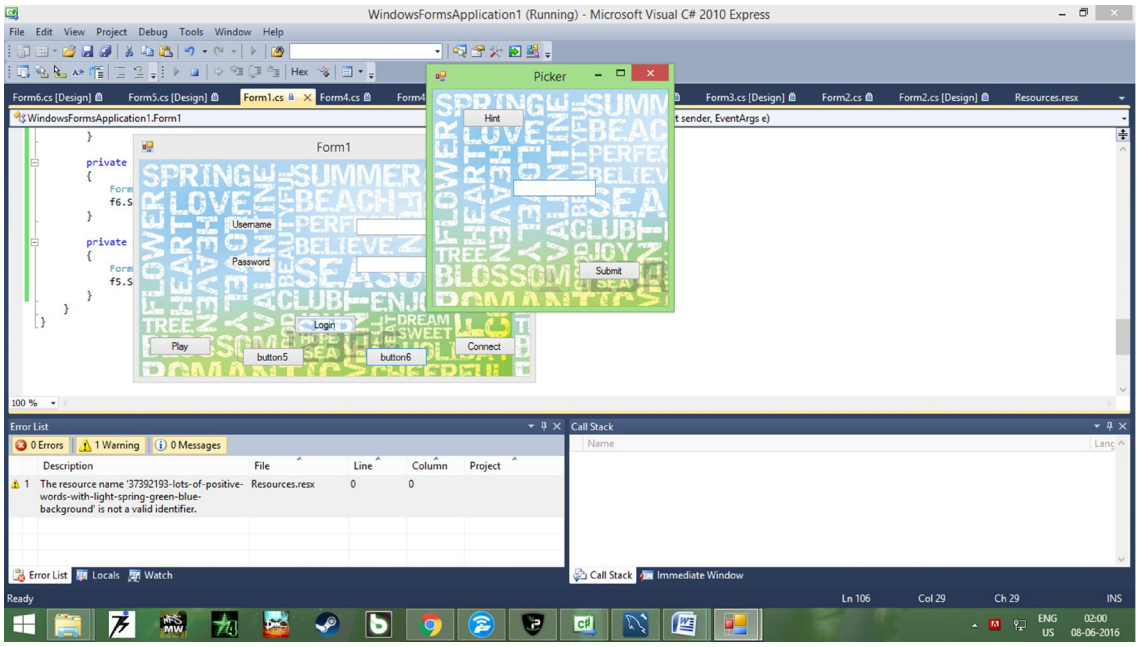

WindowsFormsApplication1 (Running) - Microsoft Visual C# 2010 Express
Picker
Hint
Submit
Form1
Username
Password
Login
Play
button5
button6
Connect
Error List
0 Errors
1 Warning
0 Messages
Description
File
Line
Column
Project
The resource name '37392193-lots-of-positive-words-with-light-spring-green-blue-background' is not a valid identifier.
Resources.resx
Call Stack
Immediate Window
Ready

Fig 4.5

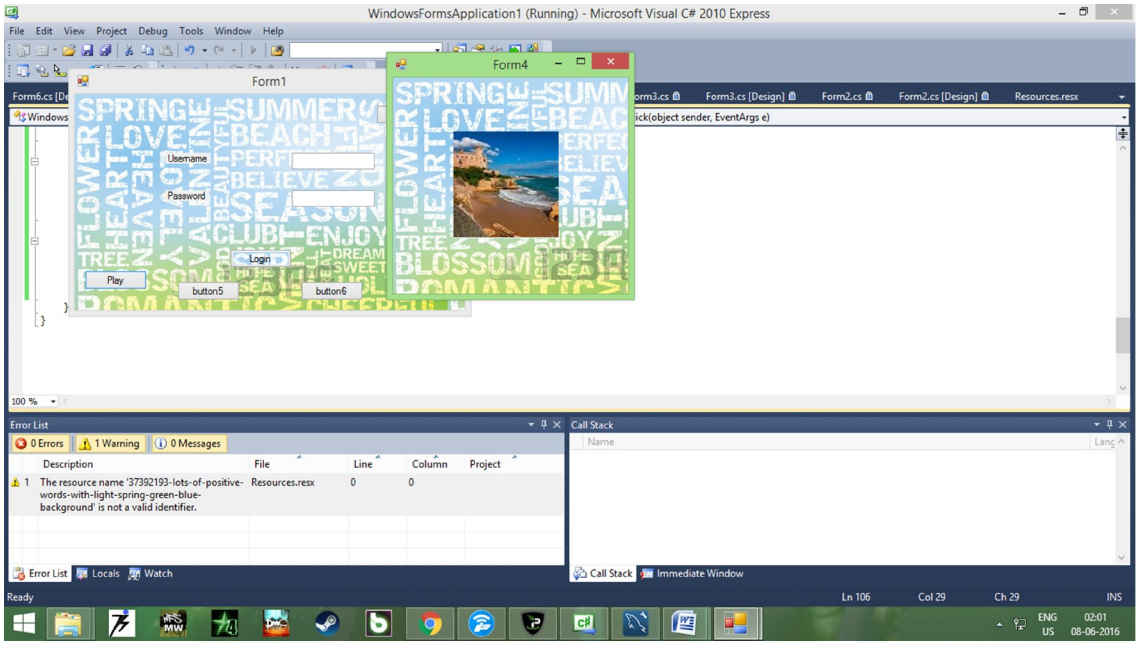

WindowsFormsApplication1 (Running) - Microsoft Visual C# 2010 Express
Form4
Form1
Username
Password
Login
Play
button5
button6
Error List
0 Errors
1 Warning
0 Messages
Description
File
Line
Column
Project
The resource name '37392193-lots-of-positive-words-with-light-spring-green-blue-background' is not a valid identifier.
Resources.resx
Call Stack
Immediate Window
Ready

Fig 4.6

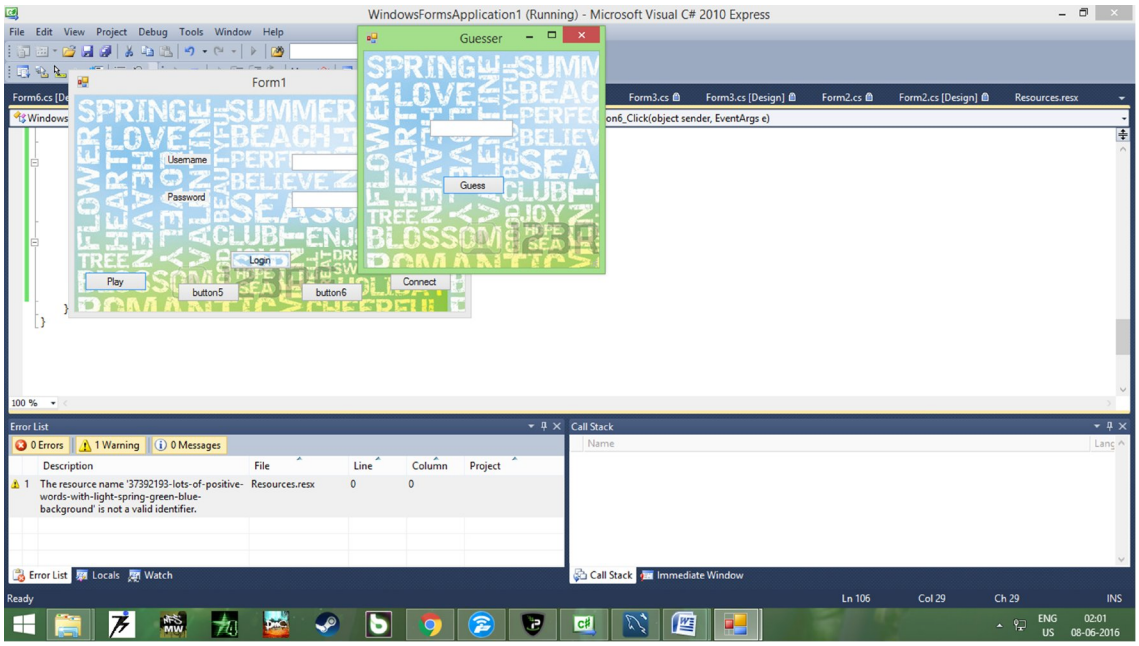

WindowsFormsApplication1 (Running) - Microsoft Visual C# 2010 Express
File Edit View Project Debug Tools Window Help
Guesser
Guess
Form1
Username
Password
Login
Play
button5
button6
Connect
Form3.cs
Form3.cs [Design]
Form2.cs
Form2.cs [Design]
Resources.resx
100 %
Error List
0 Errors
1 Warning
0 Messages
Description
File
Line
Column
Project
1 The resource name '37392193-lots-of-positive-words-with-light-spring-green-blue-background' is not a valid identifier.
Resources.resx
0
0
Error List
Locals
Watch
Call Stack
Name
Call Stack
Immediate Window
Ready
Ln 106
Col 29
Ch 29
INS
ENG
US
02:01
08-06-2016

Fig 4.7

# CHAPTER 5

## 5.1 Future Works:

### Image Search on the Web

Finding effective methods to search and retrieve images on the Web has been a prevalent line of research, both academically and in industry . Text -based image retrieval systems such as Google web search annotate images with text derived from the HTML documents that display them. The text can include the caption of the image, text surrounding the image, the entire text of the containing page, the filename of the containing HTML document, and the filename of the image itself. More recent proposals such as google web search also make use of the link structure of the Web to assign “authority” values to the images. Images that come from more authoritative web pages (e.g., pages with higher Page Rank) are displayed before images coming from less authoritative pages. This improves the quality of the results by typically showing more relevant images first.

Another possibility that has been explored involves combining text -based systems with computer vision techniques as in we are trying to do now in this project. This approach allows different types of queries to be processed (e.g., similarity queries), but doesn’t imply a significant improvement over the other approaches when it comes to standard text -based queries. The fundamental limitation of current methods for image retrieval on the Web is the heavy use of text to determine the contents of images. Text adjacent to the images is often scarce, and can be misleading or hard to process .

Because of this, many queries return inappropriate results. Figure 5, for instance, illustrates an example of Google Image Search returning a picture of a map of Chicago as the first result on the query “car.”

Fig 5.1

We argue that our game can improve the quality of image retrieval systems by providing meaningful labels that are independent of the text contained in the web pages.

## Using Our Labels

This game is primarily concerned with obtaining appropriate labels for images, and not with how these labels should be used. In the case of image search, building the labels into the current systems is not difficult, since they can be thought of as HTML captions or text appearing right next to the image. This naïve strategy would already signify an improvement over the current techniques, as these captions would provide more useful data to work with. More intelligent techniques could be conceived, such as assigning a higher weight to labels coming from the ESP game as opposed to regular HTML captions, or a numerical weight based on the "good label threshold". However, arriving at an optimal strategy for using the labels is outside the scope of this paper and is left as future work. In the case of providing textual descriptions for the visually impaired, using the labels is slightly less trivial. Our game produces labels, not explanatory sentences. While keyword labels are perfect for certain applications such as image search, other applications such as accessibility would benefit more from explanatory sentences. Nevertheless, having meaningful labels associated to images for accessibility purposes is certainly better than having nothing. Today's screen-reading programs for the visually impaired use only image filenames and HTML captions when attempting to describe images on the Web — the majority of images on the Web, however, have no captions or have non-descriptive filenames [14]. We propose that all the labels collected using the game be available for use with screen readers and that users determine themselves how many labels they want to hear for every image. Again, extensive tests are required to determine the optimal strategy.

## 5.2 Conclusion:

"Hunch Men" is a novel, complete game architecture for collecting image metadata. Segmenting objects in images is a unique challenge, and we have tailored a game specifically to this end. In the very near future, we would like to make our 1,000,000+ pieces of data available to the world by formatting it as an image segmentation library.

Like the ESP Game and Peekaboom, "Hunch Men" encompasses much more than just a simple game delivered from a website. Rather, the ideas behind the design and implementation of the game generalize to a way of harnessing and directing the power of the most intricate computing device in the world — the human mind. Some day computers will be able to segment objects in images unassisted, but that day is not today.

Today we have engines like "" that use the wisdom of humans to help naïve computers get to that point. The actual process of making computers smarter given segmentation metadata is beyond the scope of this paper, since it would require a far more sophisticated interpretation of the data than the simple bounding box derivation we have presented. Thus, we see great potential in future work at the crossroads of human-computer interaction and artificial intelligence, where the output of our interactive system helps advance the state of the art in computer vision.

.

## References

1. **Luis von Ahn, Ruoran Liu and Manuel Blum** Computer Science Department, Carnegie Mellon University 5000 Forbes Avenue, Pittsburgh PA 15213 {biglou, royliu, mblum}@cs.cmu.edu

2. **Luis von Ahn and Laura Dabbish** School of Computer Science Carnegie Mellon University Pittsburgh, PA, USA {biglou,dabbish}@cs.cmu.edu

3. en.wikipedia.org/wiki/Computer_vision

4. en.wikipedia.org/wiki/Metadata

5. http://in.mathworks.com/discovery/object-detection.html

6. https://www.cs.cmu.edu/~biglou/Peekaboom.pdf

7. https://en.wikipedia.org/wiki/ESP_game

8. https://www.cs.cmu.edu/~biglou/ESP.pdf