Eliciting User Preferences through an Online Game

Project Report submitted in fulfillment of the requirement for the degree of

Bachelor of Technology.

in

Information Technology

Under the Supervision of

Mr. Arvind Kumar

By

Pallavi Chansoria, 111428

Jaypee University of Information Technology

Waknaghat, Solan – 173234, Himachal Pradesh

Certificate

This is to certify that project report entitled **“Eliciting User Preferences through an Online Game”**, submitted by **Pallavi Chansoria** in fulfillment for the award of degree of Bachelor of Technology in Information Technology to Jaypee University of Information Technology, Waknaghat, Solan has been carried out under my supervision.

This work has not been submitted partially or fully to any other University or Institute for the award of this or any other degree or diploma.

Date: **Arvind Kumar**

Assistant Professor

Acknowledgements

I have taken sincere efforts to accomplish this project. However, it would not have been possible without the kind support and help of many individuals. I would like to extend my sincere thanks to all of them.

I am highly indebted to my project supervisor, Mr. Arvind Kumar for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support.

I would like to express my gratitude towards my parents for their kind co-operation and encouragement which helped me.

I would like to express my special gratitude and thanks to the faculty of CS-IT for giving me such attention and time as and when I had any questions for them.

Date: Pallavi Chansoria

Table of Contents

S.No. **Topic** **Page No.**

List of Figures

ABSTRACT

The question of ‘Who has created us?’ has always bothered mankind. There have been, essentially, two sets of beliefs- One believing in a higher power that put man as he is today, on this planet. The other, is the theory of evolution.
In both these cases, one thing is common- development i.e. where we started and where we are today. In this process of development, one thing has always captured the imagination of man- Genesis. The need to answer the question of a creator and to be able to witness ‘man-made’ life has been a dream passed on through generations.

Machines come into play here. Machines are the result of this need to create. Machines have evolved along these years. One thing machines have not been able to achieve yet, is consciousness. Although, we have succeeded in providing decision-making capabilities to machines but not completely, not in the terms of awareness or understanding. However, machine-learning is a fast growing field.

This project deals with the field of Human Computation- a method employed to solve problems that neither the computer nor the man can solve alone. It highlights a new method to elicit user preferences that does not ask users to tell what they prefer, but rather what a random person would prefer, and rewards them if their prediction is correct. We provide an implementation of our method as a two-player game in which each player is shown two images and asked to click on the image their partner would prefer. Both these players are chosen completely at random from among those that are online and if both partners click on the same image, they both obtain points, whereas if they click on different images, neither of them receives any points.

The Game comes under the category of GWAP- Games with a Purpose. We compare several algorithms for combining these relative judgments between pairs of images into a total ordering of all images. In addition, we show how merely observing user preferences on a specially chosen set of images can predict a user‘s gender with high probability.

CHAPTER 1 INTRODUCTION

The online game comes under the category of ‘GWAP’ i.e. Games with a Purpose. A **human-based computation game** or **game with a purpose** is a human-based computation technique in which a computational process performs its function by outsourcing certain steps to humans in an entertaining way. This approach uses differences in abilities and alternative costs between humans and computer agents to achieve symbiotic human–computer interaction. These tasks can include labeling images to improve web searching, transcription of ancient text and any activity requiring common sense or human experience. There are things humans cannot do alone and there are things computers cannot do alone; Human computation focuses on bringing the two together to achieve a solution, example: CAPTCHA. Recently, video games with a purpose have been proposed to lower the cost of annotations and increase the level of player's engagement. The examples are endless- Apetopia, Artigo, ESP Game.

We provide an implementation of our method as a two-player game in which each player is shown two images and asked to click on the image their partner would prefer. Both these players are chosen completely at random from among those that are online and if both partners click on the same image, they both obtain points, whereas if they click on different images, neither of them receives any points. We show how merely observing user preferences on a specially chosen set of images can predict a user’s gender with high probability.

1.1 Purpose of the Game

We use the players’ preferences between images to create a simple gender model. Based on only ten pair-wise judgments, our model can determine a player’s gender with high probability. This shows that responding to a request for seemingly benign information, such as which of two images a user prefers, can actually reveal significant information about a person. Under these circumstances, it becomes questionable whether people really can protect their privacy online.

Chapter 2 LITERATURE REVIEW

2.1 Early Work in Human Computation

Human-based computation (apart from the historical meaning of "computer") research has its origins in the early work on interactive evolutionary computation. The idea behind interactive evolutionary algorithms is due to Richard Dawkins. In the Biomorphs software accompanying his book *The Blind Watchmaker* (Dawkins, 1986) the preference of a human experimenter is used to guide the evolution of two-dimensional sets of line segments. In essence, this program asks a human to be the fitness function of an evolutionary algorithm, so that the algorithm can use human visual perception and aesthetic judgment to do something that a normal evolutionary algorithm cannot do. However, it is difficult to get enough evaluations from a single human if we want to evolve more complex shapes. Victor Johnston and Karl Sims extended this concept by harnessing the power of many people for fitness evaluation (Caldwell and Johnston, 1991; Sims, 1991). As a result, their programs could evolve beautiful faces and pieces of art appealing to public. These programs effectively reversed the common interaction between computers and humans. In these programs, the computer is no longer an agent of its user, but instead, a coordinator aggregating efforts of many human evaluators. These and other similar research efforts became the topic of research in aesthetic selection or interactive evolutionary computation (Takagi, 2001), however the scope of this research was limited to outsourcing evaluation and, as a result, it was not fully exploring the full potential of the outsourcing.

A concept of the automatic Turing test pioneered by Moni Naor (1996) is another precursor of human-based computation. In Naor's test, the machine can control the access of humans and computers to a service by challenging them with a natural language processing (NLP) or computer vision (CV) problem to identify humans among them. The set of problems is chosen in a way that they have no algorithmic solution that is both effective and efficient at the moment. If it existed, such an algorithm could be easily performed by a computer, thus defeating the test. In fact, Moni Naor was modest by calling this an automated Turing test. The Imitation Game described by Alan Turing (1950) didn't propose using CV problems. It was only proposing a specific NLP task, while the Naor test identifies and explores a large class of problems, not necessarily from the domain of NLP that could be used for the same purpose in both automated and non-automated versions of the test.

Finally, Human-based genetic algorithm (HBGA) encourages human participation in multiple different roles. Humans are not limited to the role of evaluator or some other predefined role, but can choose to perform a more diverse set of tasks. In particular, they can contribute their innovative solutions into the evolutionary process, make incremental changes to existing solutions, and perform intelligent recombination. In short, HBGA allows humans to participate in all operations of a typical genetic algorithm. As a result of this, HBGA can process solutions for which there are no computational innovation operators available, for example, natural languages. Thus, HBGA obviated the need for a fixed representational scheme that was a limiting factor of both standard and interactive EC. These algorithms can also be viewed as novel forms of social organization coordinated by a computer.

2.2 Luis Von Ahn

Fig. 2.2.1

Luis von Ahn (born 1979) is a Guatemalan entrepreneur and an associate professor in the Computer Science Department at Carnegie Mellon University. He is known as one of the pioneers of crowd sourcing. He is the founder of the company reCAPTCHA, which was sold to Google in 2009, and the co-founder and CEO of Duolingo, a popular language-learning platform.

As a professor, his research includes CAPTCHAs and human computation, which has earned him international recognition and numerous honours.

Von Ahn's early research was in the field of cryptography. With Nicholas J. Hopper and John Langford, he was the first to provide rigorous definitions of steganography and to prove that private-key steganography is possible.

In 2000, he did early pioneering work with Manuel Blum on CAPTCHAs, computer-generated tests that humans are routinely able to pass but that computers have not yet mastered. These devices are used by web sites to prevent automated programs, or bots, from perpetrating large-scale abuse, such as automatically registering for large numbers of accounts or purchasing huge number of tickets for resale by scalpers.

Von Ahn's Ph.D. thesis, completed in 2005, was the first publication to use the term "human computation" that he had coined for methods that combine human brainpower with computers to solve problems that neither could solve alone. Von Ahn's Ph.D. thesis is also the first work on Games With A Purpose, or GWAPs, which are games played by humans that produce useful computation as a side-effect. The most famous example is the ESP Game, an online game in which two randomly paired people are simultaneously shown the same picture, with no way to communicate. Each then lists a number of words or phrases that describe the picture within a time limit, and are rewarded with points for a match. This match turns out to be an accurate description of the picture, and can be successfully used in a database for more accurate image search technology. The ESP Game was licensed by Google in the form of the Google Image Labeller, and is used to improve the accuracy of the Google Image Search. Von Ahn's games brought him further coverage in the mainstream media. His thesis won the Best Doctoral Dissertation

In 2007, von Ahn invented reCAPTCHA, a new form of CAPTCHA that also helps digitize books. In reCAPTCHA, the images of words displayed to the user come directly from old books that are being digitized; they are words that optical character recognition could not identify and are sent to people throughout the web to be identified. ReCAPTCHA is currently in use by over 100,000 websites and is transcribing over 40 million words per day.

As of 2014, von Ahn is working on Duolingo, a company that aims to coordinate millions of people to translate the Web into every major language.

2.3 Taxonomy of Methods

2.3.1 Absolute versus Relative Judgments

First, we make a distinction between absolute and relative judgments. An absolute judgment is a judgment that assigns an absolute score to an item, such as a star rating from 1-5 where 1 is worst and 5 is best. On the other hand, relative judgment only compares items, i.e., ―this image is better than that image.

Absolute ratings have two problems: *calibration* (or better, the lack of calibration) and *limited resolution*. Calibration is the problem of defining what a particular rating means compared with previous ratings and compared with other people's ratings. For example, if I usually assign 1 or 2, my 4 might have the same meaning as someone else's 5. Also, judgment may change during the rating process: for example, a user might in their first rating give a 5 to a good image, only to discover later that there are far better images. Thus, they may want to change their first rating to a 4.

This creates systematic errors in the data. Limited resolution is the problem of assigning a rating to an image that is only marginally better than a different image. Assuming that the rating system only has 5 levels, the user might give it the same rating even though they clearly think it is better (but not good enough for the next level). In this case we lose information. To overcome the problem of limited resolution, one could simply use a rating system with finer granularity, say from 1 to 100. However, many judges will not adapt to this system, but instead keep a scale of 1 to 10 in their mind and map 8 to 80 and so forth. Relative judgments have the advantage that they are usually easy to make. In most cases, they do not change after seeing new information, i.e., a user who prefers image A over image B will still do so after they have seen other images. Even if the absolute ratings of image A and image B change over time, their relative ordering is likely to stay the same. Therefore, old absolute ratings are more likely to be inaccurate than old relative ratings.

2.3.2 Total versus Partial Judgments

By total judgments, we mean that the judges are required to make judgments about all of the images. In the case of absolute ratings, the user is required to rate all n images. In the case of relative ratings, the user is required to compare every image with every other image, which is on the order of $O(n^2)$ comparisons. Total judgments are, therefore, infeasible for large datasets. Partial judgments, however, have the problem of how to deal with incomplete data.

2.3.3 Random Access versus Predefined Access

By random access, we mean that the judges are allowed to search for particular items and then rate them. This has the advantage that the judges can focus on rating things in which they are most interested. However, it has a major drawback: it opens the door to malicious manipulation. Judges could easily search for their own pictures and always give them the highest ratings. This behaviour cannot easily be stopped on the Internet since the cost of creating new fake identities is extremely low and it is not (currently) possible to tell fake accounts from real accounts. Another drawback of random access methods is that some items receive many ratings while others receive few. In such cases, combining the ratings becomes difficult.

By predefined access, we mean methods where the users are given images to rate in a *predefined sequence*. Thus, the users cannot influence which images they can rate. While theoretically it is still possible to cheat just by waiting for one's own images, it is much harder. In a method employing random access, the chance of being able to rate one's own images is 1. For a method that randomly shows one out of n images (with replacement) to rate, the chance of being able to rate one's own image is $1/n$, and the expected time to wait until one sees their own image is n. Therefore, methods that use predefined access have the desirable property that the possibility of cheating decreases as the amount of data increases.

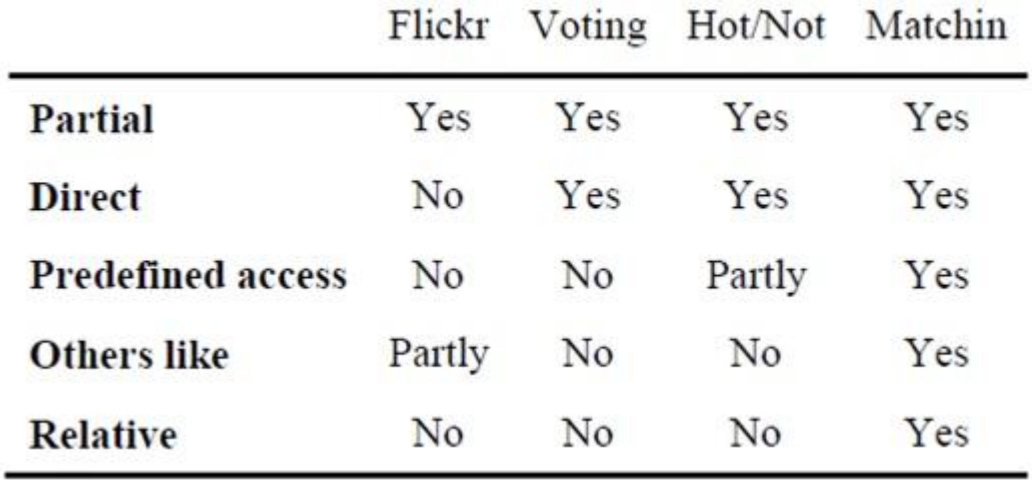
2.3.4 "I Like" Versus "Others Like"

Another important distinction between methods is whether the judges are asked —what do you like? Or what do you think others will like? Although the difference looks subtle at first, it has major implications. We can compare this to the problem of predicting elections. The most common way is to poll potential voters and ask them who they would vote for in the upcoming election. One then takes the sample average as an estimate of the future election result. This is the ‘I like’ case. The other option is to ask them, who do they think is going to win the election? In this case they will not only consider their own opinion but also the opinion of their friends and relatives in combination with external information (polls, news, etc.). This is the ‘others like’ case.

In the ‘I like’ case, every voter automatically becomes a weak predictor, because every voter only has a limited amount of information at his/her disposal. In this ‘others like’ case we can make a further distinction between methods that ask for what one particular partner might prefer and what other people in general prefer.

2.3.5 Direct Versus Indirect

By direct, we mean methods that ask the judges about the ‘beauty’ of an image. Indirect methods would infer beauty through meta-information. Examples of meta-information are number of views, number of comments, number of tags, and number of pages linking to a particular image. Indirect methods have the disadvantage that, once the methods are known, their ratings can quickly be subjected to cheating. People could easily create many comments on their own images, add lots of tags, create dummy pages linking to their images, etc. This means that even though an indirect method might use predefined access (for example by crawling images and counting incoming links, etc.), people still have random access to the meta-information and can change it in any way they want.



	Flickr	Voting	Hot/Not	Matchin
Partial	Yes	Yes	Yes	Yes
Direct	No	Yes	Yes	Yes
Predefined access	No	No	Partly	Yes
Others like	Partly	No	No	Yes
Relative	No	No	No	Yes

Fig 2.2.2 A Taxonomy of Methods

2.4 Existing Methods

2.4.1 Flickr Interestingness

The popular online photo sharing Web site Flickr has developed its own algorithm to rank images. Although their algorithm has not been published, we know from their patent application that it is at least partly based on meta-data such as the quantity of user-entered meta-data concerning the media object, the number of users who have assigned metadata to the media object, an access pattern related to the media object, a lapse of time related to the media object, and/or; on the relevance of metadata to the media object. (We note that all these meta-data can easily be faked.)

This means that Flickr's 'interestingness' does not measure 'beauty' directly. Some of the meta-data measures how much other people will possibly like an image. A link to an image, for example, is usually created because the authors think the image might be interesting to their readers. However, the problem with all methods that rely on meta-data (like number of comments) is that established long-term users who have many friends on that network have an advantage. Ultimately, it is not clear whether 'interestingness' measures the interestingness of the image or the popularity of its author.

2.4.2 Voting

Perhaps the simplest method of eliciting user preferences is just to let users vote on images, using approval/ disapproval or a rating scale (e.g., 1 to 5 stars). Users can search for particular items and vote on them (random access). This is possibly the most frequently used method on the Web: Digg, YouTube and others all use variants of this scheme to rate and rank their content. These methods, since they are based on random access, share the common characteristic that some items receive many votes while others receive few. This leads to a new problem of combining these ratings into a global ranking. If two items have the same average rating, but one has 1,000 votes while the other one only has 10 votes, the one with more votes should probably be ranked higher. However, generalizing this principle is non- trivial.

2.4.3 Hot or Not

The Internet site 'Hot or Not' uses a voting system from 1 to 10. It is limited in that it ranks only images of people. The most important difference from the previously mentioned sites is that a normal user is given random images to rate; they cannot search for them. However, it is still possible for people to send a link to an image to a friend who can then rate the picture. Therefore, it is still easy for malicious users to cheat and rate their friends’ pictures higher than they might rank otherwise.

Chapter 3 THE GAME

3.1 Mechanism

It is a two-player game that is played over the Internet. At the beginning of the game, a player is matched randomly with another person who is visiting the game's Web site at the same time. If there is no other player available at the same time, we pair them with a bot (a computer that plays as if it was a human). After the player is matched with its partner (either human or machine), they play several rounds. In each round, the two players see the same two images and both are asked to click on the image that they think their partner likes better. If they agree, they both receive points. Thus, if the players want to score many points, they not only have to consider which image they prefer, but also which image their partner might prefer. To make the game more fun, the players are given more points for consecutive agreements. More specifically, the game uses a sigmoid function for scoring games. The scoring function is shown in the figure below.

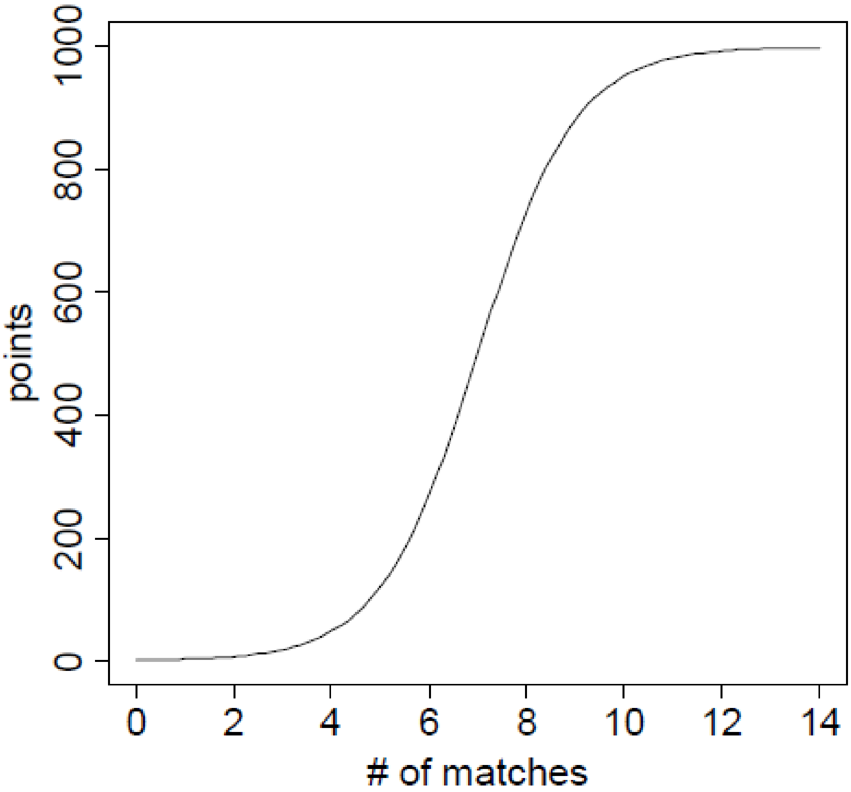


Fig. 3.1.1 Sigmoid Scoring Function

While the first match only earns a few points, the second and third match in a row earn exponentially more points until the seventh match at which point the growth of the function decreases. At the end of the game, the two players can review all of their decisions and chat with each other. All clicks are recorded and stored in a database. We also store the time it took for the users to make a decision.

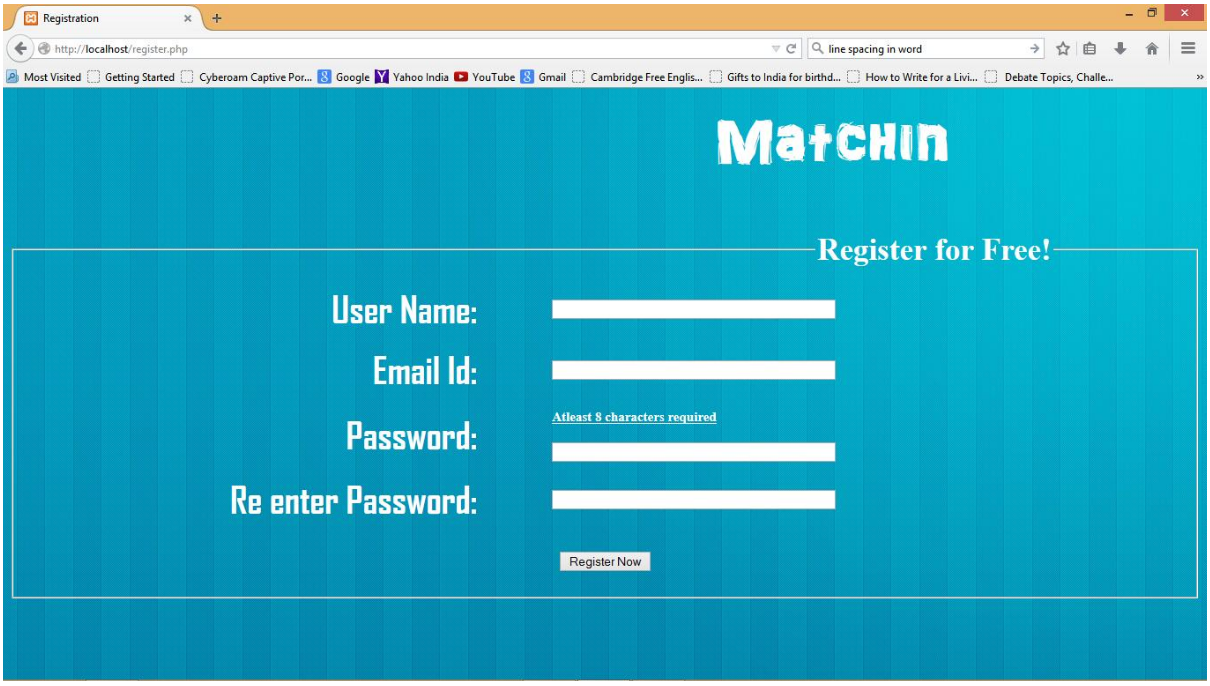
The bot uses these stored clicks to emulate a human as closely as possible. When it sees two images, it clicks on the image that was considered to be better by a human in an earlier game. The bot mimics the same person for the entire game. Also, the bot waits exactly as long as the human did. (Note, however, that the bot's clicks are not recorded.) In every round, we show two random pictures from our collection of images, favouring images for which we have not yet collected enough data. The images were collected from different sources. Each one was handpicked.

One of the underlying design objectives of the game is to remove any systematic errors in the resulting data, i.e., all of the judgments should be correct in that they truly reflect the judge's opinion. The judgments should also be robust in the sense that they should still be considered valid after a long time. It is thus a rating system that uses the previously defined concepts of relative judgment and predefined access. It gives an incentive for the judges to consider not only their own opinion, but also the possible opinions of others in making their judgments. Because of predefined access, the game is very hard to cheat because the user can only compare images and the impact of a single malicious judgment is minimal. We also note that to minimize cheating, images are presented to both players in random order (i.e., the image on the left for one player might be the image on the right for the other player).

The main difficulty in creating games with a purpose is to make them enjoyable and an exciting scoring function does the trick. If the game used a constant scoring function, say 100 points for every agreement, players could get many points by quickly picking the images at random (e.g., always choosing the one on the left). This allowed players to get 100 points with 50% probability in every round even without taking the time to look at every image and thus made the game less enjoyable. A scoring function that gave a higher reward to consecutive agreements was needed. A linear scoring function where the first match earned 100 points; the second earned 200, the third 300, etc could be used. This made the game more fun. An exponential scoring function is even more exciting. The rewards, however, can become too high so a limit on the amount of points that could be earned in a single round was needed. That is why we chose the sigmoid function.

3.2 Project Design

3.2.1 Registration



MatcHin

Register for Free!

User Name:

Email Id:

Atleast 8 characters required

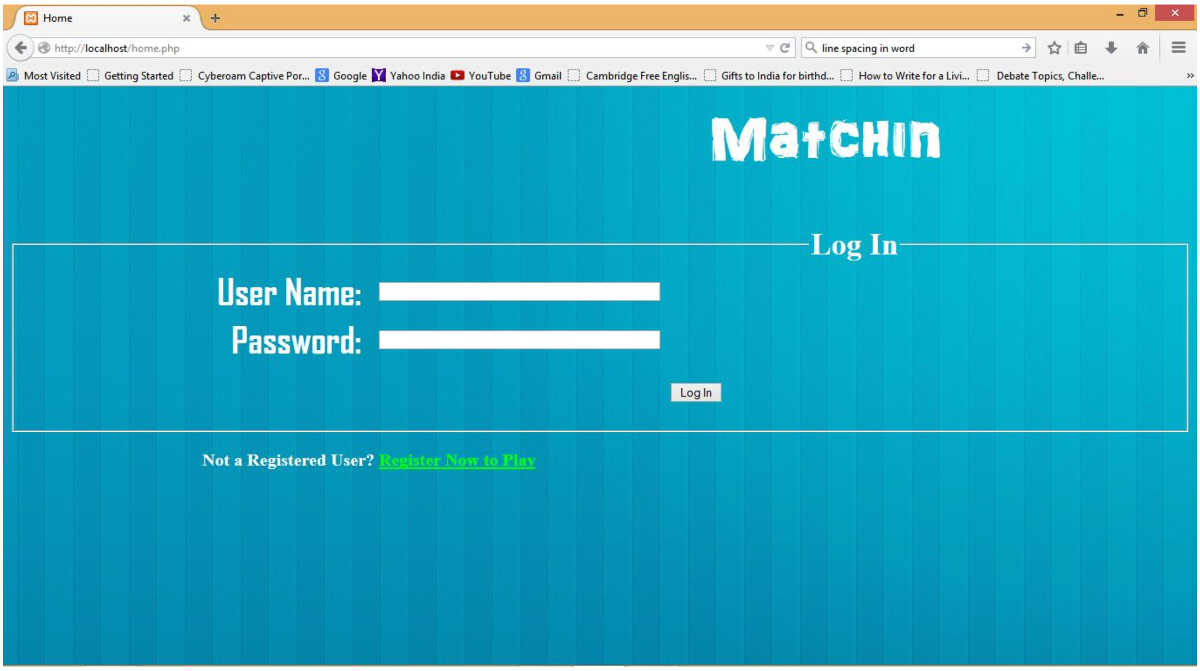
Password:

Re enter Password:

Register Now

Fig. 3.2.1 The Registration Page

3.2.2 Login



MatcHin

Log In

User Name:

Password:

Log In

Not a Registered User? Register Now to Play

Fig. 3.2.2 The Login Page

3.2.3 Start (session start)

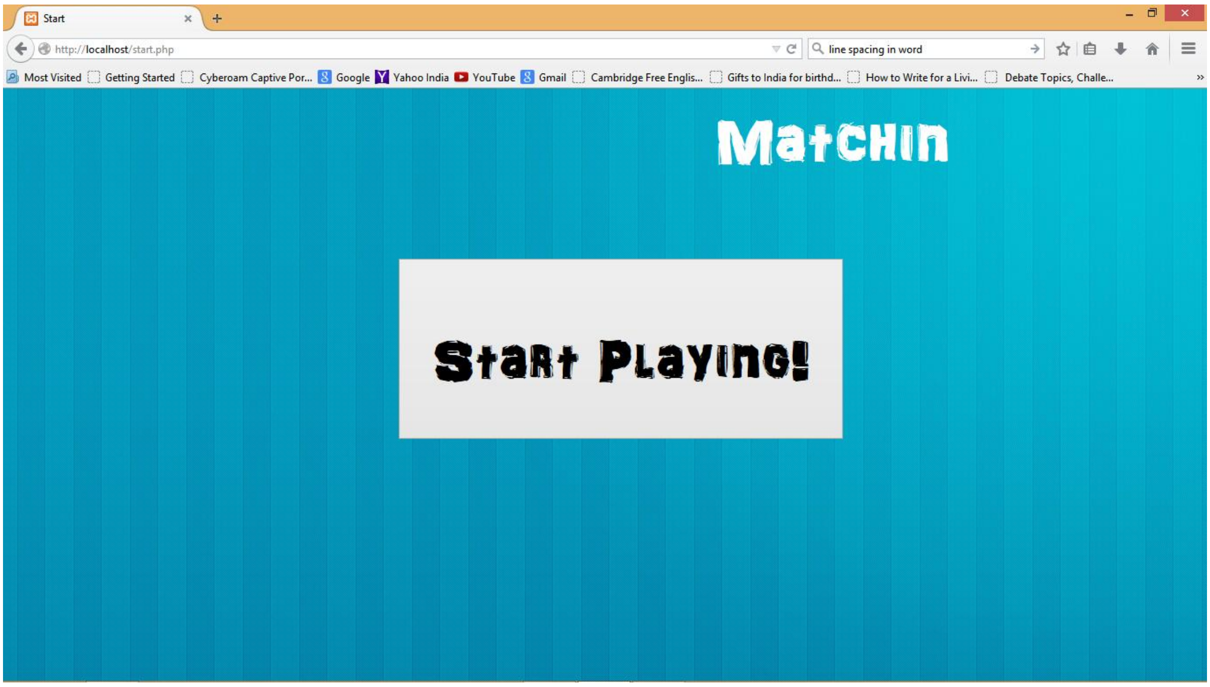
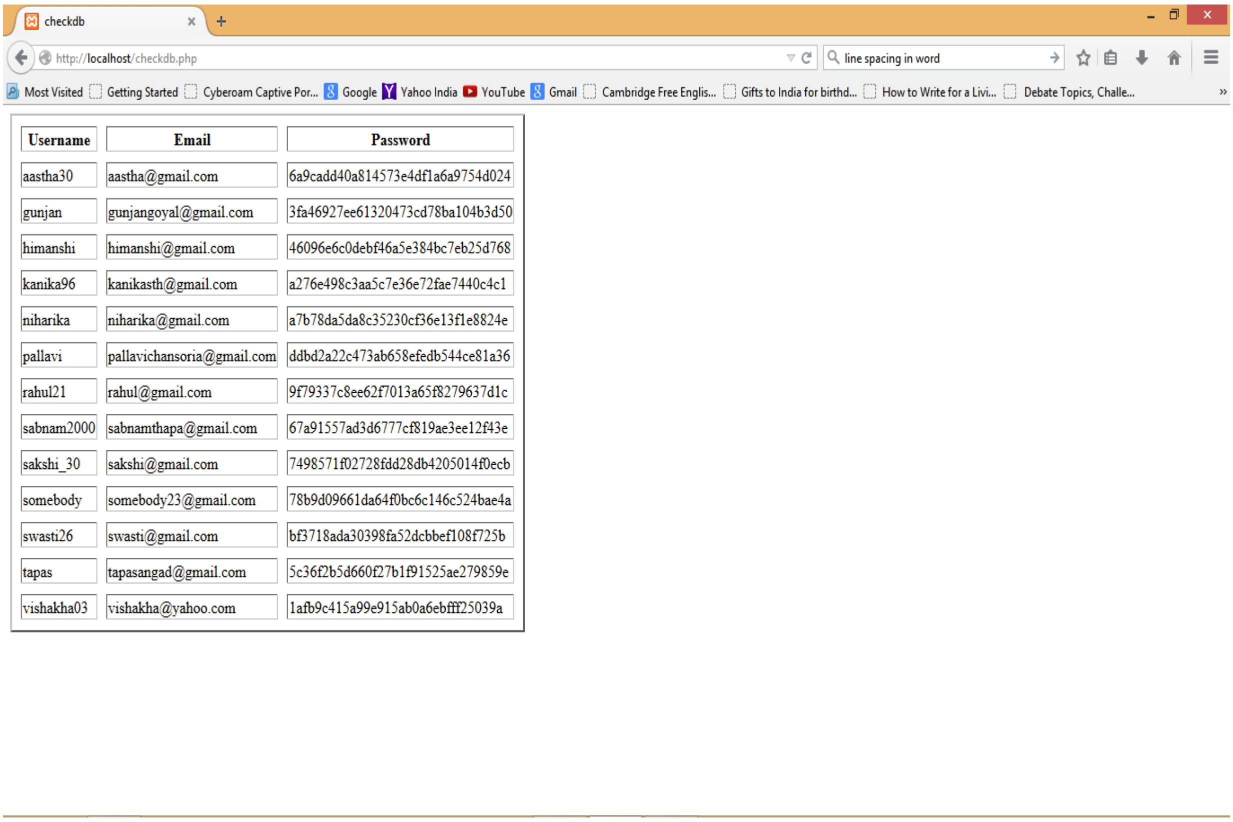


Fig. 3.2.3 The ‘Start Game’ Page

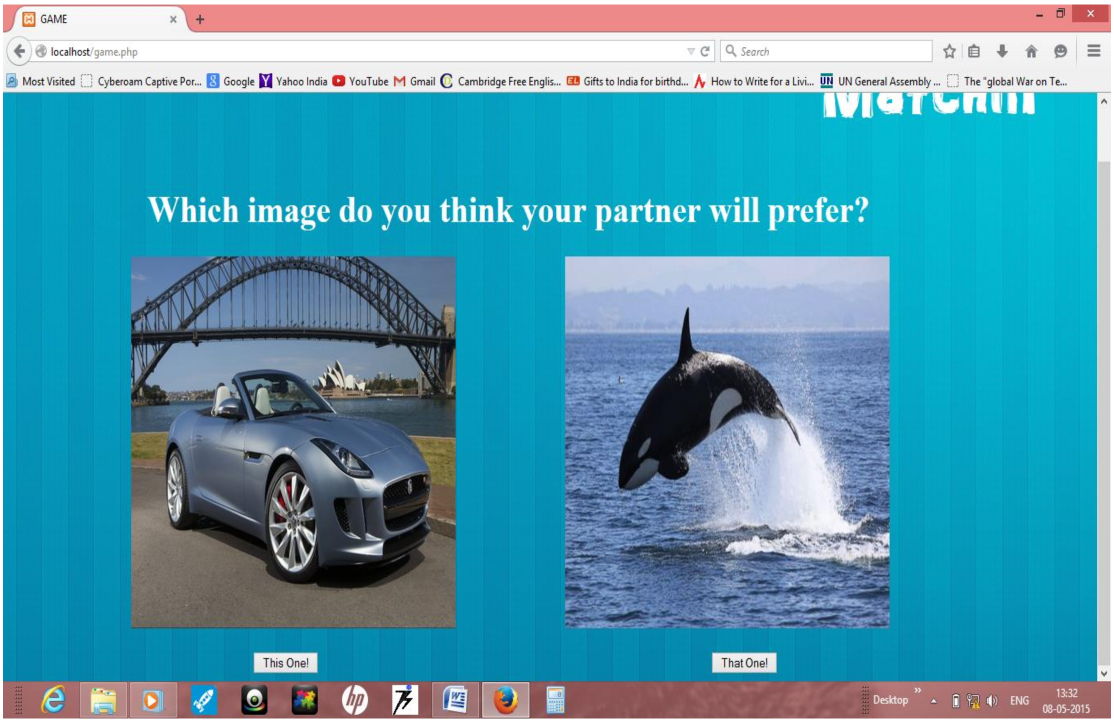
3.2.4 Registered User Database



Username	Email	Password
aastha30	aastha@gmail.com	6a9cadd40a814573e4df1a6a9754d024
gunjan	gunjangoyal@gmail.com	3fa46927ee61320473cd78ba104b3d50
himanshi	himanshi@gmail.com	46096e6c0debf46a5e384bc7eb25d768
kanika96	kanikasth@gmail.com	a276e498c3aa5c7e36e72fae7440c4c1
niharika	niharika@gmail.com	a7b78da5da8c35230cf36e13f1e8824e
pallavi	pallavichansoria@gmail.com	ddbd2a22c473ab658efedb544ce81a36
rahul21	rahul@gmail.com	9f79337c8ee62f7013a65f8279637d1c
sabnam2000	sabnamthapa@gmail.com	67a91557ad3d6777cf819ae3ee12f43e
sakshi_30	sakshi@gmail.com	7498571f02728fdd28db4205014f0ecb
somebody	somebody23@gmail.com	78b9d09661da64f0bc6c146c524bae4a
swasti26	swasti@gmail.com	bf3718ada30398fa52dcbbef108f725b
tapas	tapasangad@gmail.com	5c36f2b5d660f27b1f91525ae279859e
vishakha03	vishakha@yahoo.com	1afb9c415a99e915ab0a6ebfff25039a

Fig. 3.2.4 The Registered Users Database

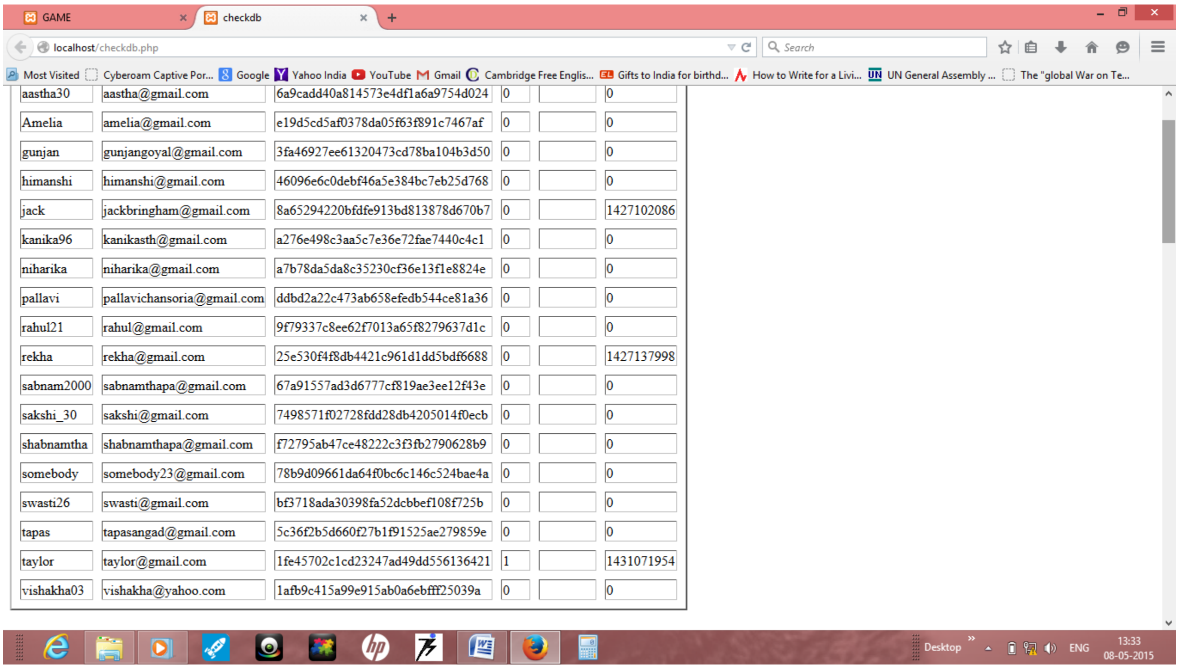
3.2.5 Game Page



Which image do you think your partner will prefer?

Fig. 3.2.5 The Game Page

3.2.6 Updated User Database (tracking session)



aastha30	aastha@gmail.com	6a9cadd40a814573e4df1a6a9754d024	0		0
Amelia	amelia@gmail.com	e19d5cd5af0378da05f63f891c7467af	0		0
gunjan	gunjangoyal@gmail.com	3fa46927ee61320473cd78ba104b3d50	0		0
himanshi	himanshi@gmail.com	46096e6c0debf46a5e384bc7eb25d768	0		0
jack	jackbringham@gmail.com	8a65294220bfdfe913bd813878d670b7	0		1427102086
kanika96	kanikasth@gmail.com	a276e498c3aa5c7e36e72fae7440c4c1	0		0
niharika	niharika@gmail.com	a7b78da5da8c35230cf36e13f1e8824e	0		0
pallavi	pallavichansoria@gmail.com	ddbd2a22c473ab658efedb544ce81a36	0		0
rahul21	rahul@gmail.com	9f79337c8ee62f7013a65f8279637d1c	0		0
rekha	rekha@gmail.com	25e530f4f8db4421c961d1dd5bdf6688	0		1427137998
sabnam2000	sabnamthapa@gmail.com	67a91557ad3d6777cf819ae3ee12f43e	0		0
sakshi_30	sakshi@gmail.com	7498571f02728fdd28db4205014f0ecb	0		0
shabnamtha	shabnamthapa@gmail.com	f72795ab47ce48222c3f3fb2790628b9	0		0
somebody	somebody23@gmail.com	78b9d09661da64f0bc6c146c524bae4a	0		0
swasti26	swasti@gmail.com	bf3718ada30398fa52dcbbef108f725b	0		0
tapas	tapasangad@gmail.com	5c36f2b5d660f27b1f91525ae279859e	0		0
taylor	taylor@gmail.com	1fe45702c1cd23247ad49dd556136421	1		1431071954
vishakha03	vishakha@yahoo.com	1afb9c415a99e915ab0a6ebfff25039a	0		0

Fig. 3.2.6 The Updated User Database Page

3.2.7 Image Database

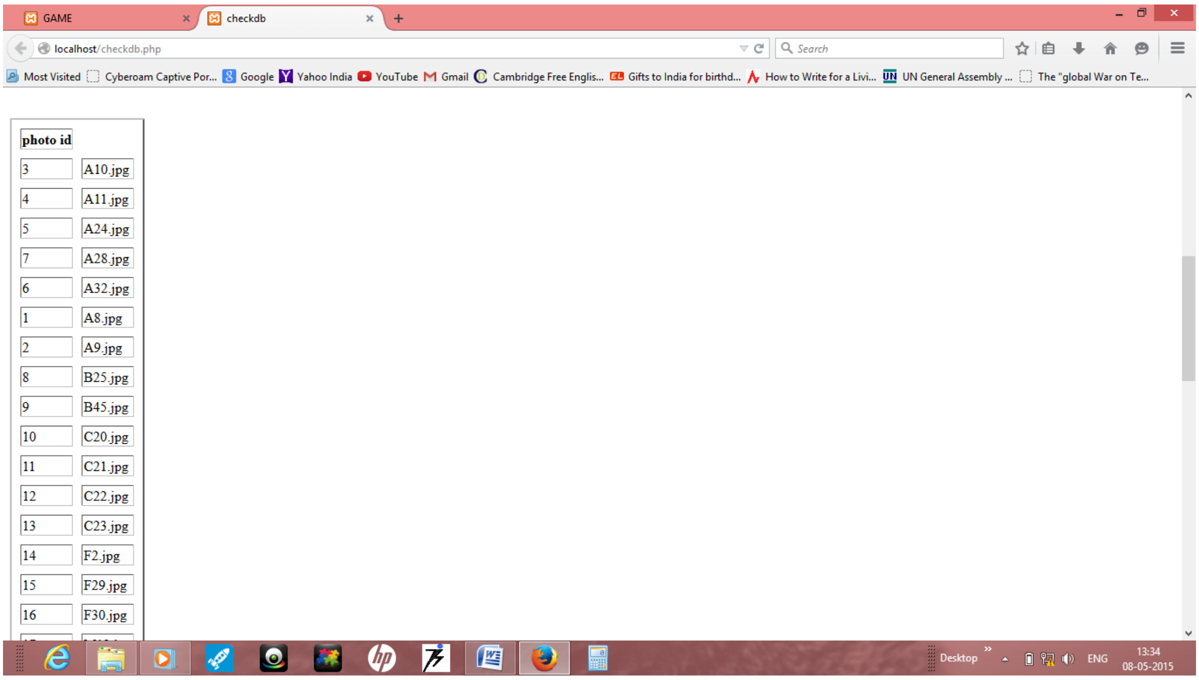


photo id	
3	A10.jpg
4	A11.jpg
5	A24.jpg
7	A28.jpg
6	A32.jpg
1	A8.jpg
2	A9.jpg
8	B25.jpg
9	B45.jpg
10	C20.jpg
11	C21.jpg
12	C22.jpg
13	C23.jpg
14	F2.jpg
15	F29.jpg
16	F30.jpg

Fig. 3.2.7 The Image Database

3.2.8 Images

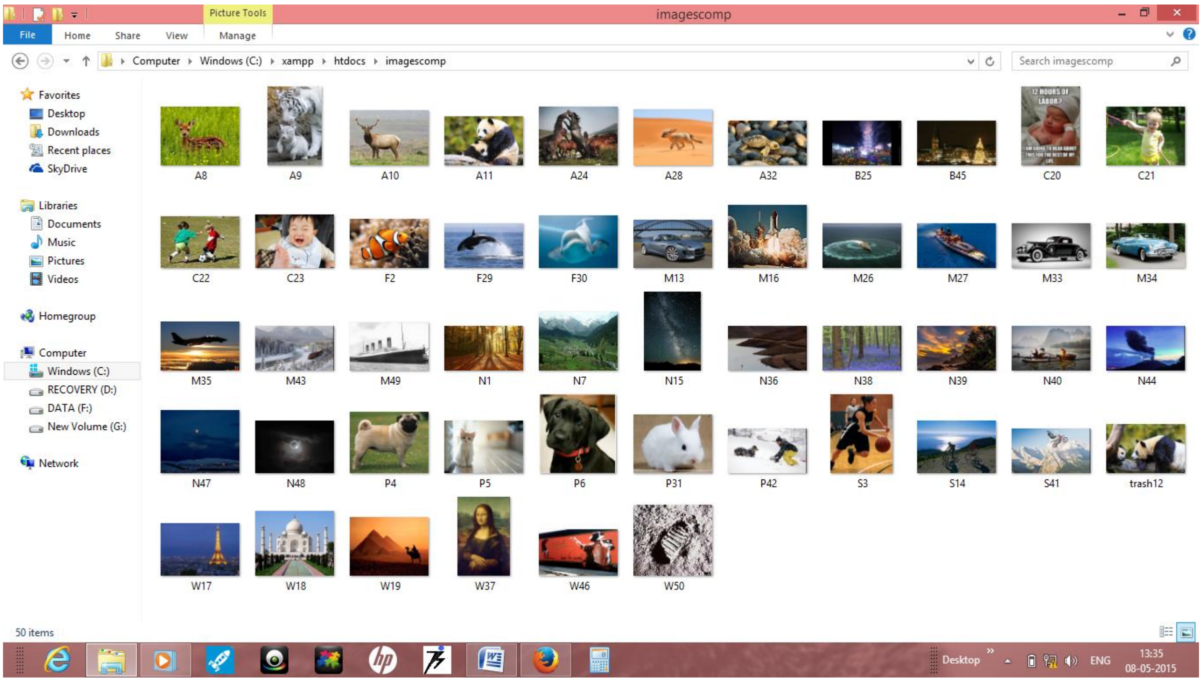


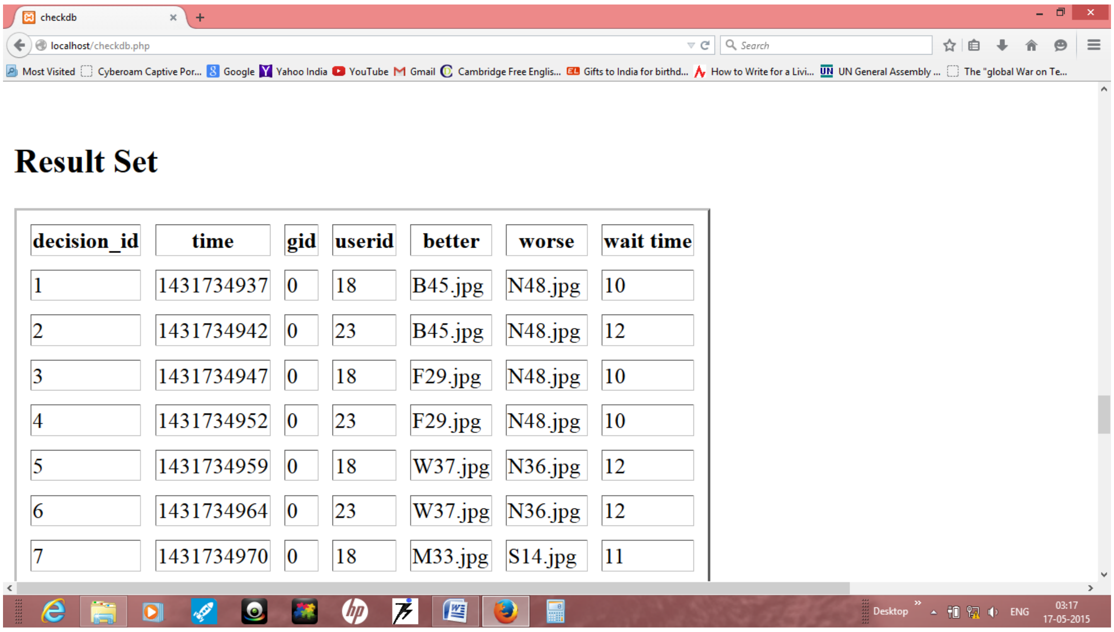
Fig. 3.2.8 All Images

3.3 The Data

An individual decision/record is stored in the form:

<id, game_id, player, better, worse, time, waiting_time>

Where *id* is a number assigned to identify the decision, *game_id* is the ID of the game, *player* is the ID of the player who made the decision in this record, *better* is the ID of the image the player considered better, *worse* is the ID of the other image, *time* is the date and time when the decision was made, and *waiting_time* is the amount of time the player looked at the two images before making a decision.



Result Set

decision_id	time	gid	userid	better	worse	wait time
1	1431734937	0	18	B45.jpg	N48.jpg	10
2	1431734942	0	23	B45.jpg	N48.jpg	12
3	1431734947	0	18	F29.jpg	N48.jpg	10
4	1431734952	0	23	F29.jpg	N48.jpg	10
5	1431734959	0	18	W37.jpg	N36.jpg	12
6	1431734964	0	23	W37.jpg	N36.jpg	12
7	1431734970	0	18	M33.jpg	S14.jpg	11

Fig. 3.3.1

3.4 Gender Prediction

The decisions of players are subjective and we can exploit this fact to create a gender test. We know the gender for a few players from their profile settings (though this information is self-reported and therefore may be wrong). To find a pair of images (A,B) that has a strong gender bias, we compute the conditional entropy $H[G|X]$ of the player's gender G given that we know the player's decision X, where $A>B$ means that image A was considered better than image B:

$H[G|X]$ = Pr (x) $H[G|X=x]x\in$ $A<B,A>B$

A pair (A,B) has a large gender bias (and is therefore good for determining the gender of a new player) if the conditional entropy $H[G|X]$ is small, i.e., learning the decision tells us a lot about the gender. The necessary conditional probabilities Pr $(G=g|X=x)$ can be computed with Bayes’ rule given the class conditionals Pr$(X=x|G=g)$. For the class conditionals, we trained two ELO predictors, one with male players only and one with female players only. We then compute $H[G|X]$ for many pairs of images and select pairs for which $H\ G\ X$ is smaller than a fixed threshold value.

To predict the gender of new users we sample 10 edges from those with strong gender bias and we ask the users to choose the image they prefer for each pair. In order to make our intentions less obvious, we add some random image pairs. Once we know their decisions on the 10 pairs, we use a simple naïve Bayes’ classifier to predict their gender. The naïve Bayes’ classifier assumes that the individual decisions are independent given the gender and chooses the label g that maximizes the likelihood of the data:

$$\hat{g} = \mathrm{argmax}_g \Pr(G = g) \prod_{i=1}^{10} \Pr(X_i = x_i | G = g)$$

If we are able to predict the gender of a player based on their image preferences, this goes on to show how someone online can ask for seemingly innocent information but derive more out of it than is accounted for. It severely questions whether we can protect our privacy online.

3.4.1 Examples of Images with a Gender Bias

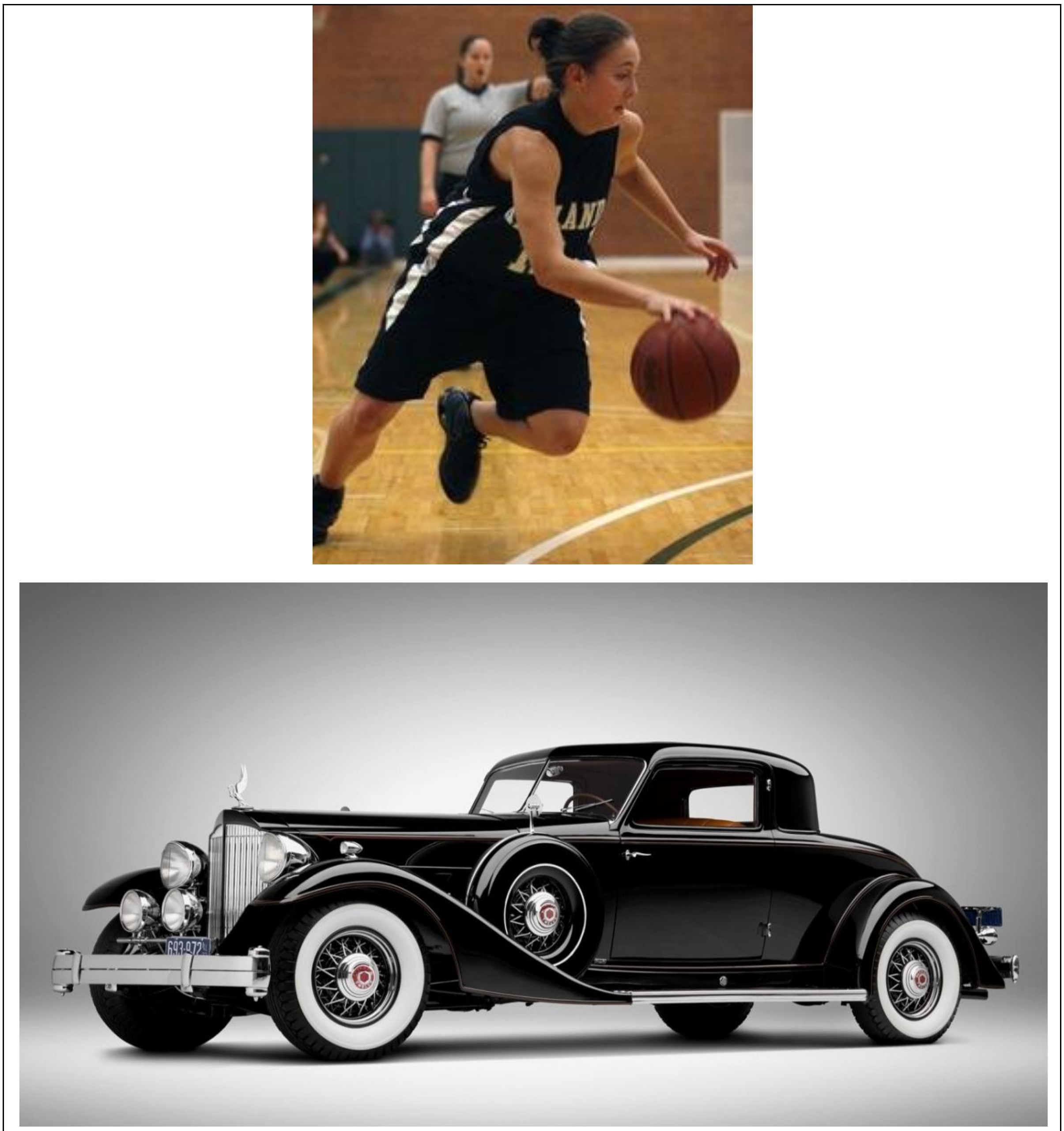


Fig.3.4.1.1

Women prefer the first image while men prefer the second in Fig. 3.4.1.1. There of course, will be exceptions but the observation remains accurate for a large percentage of players.

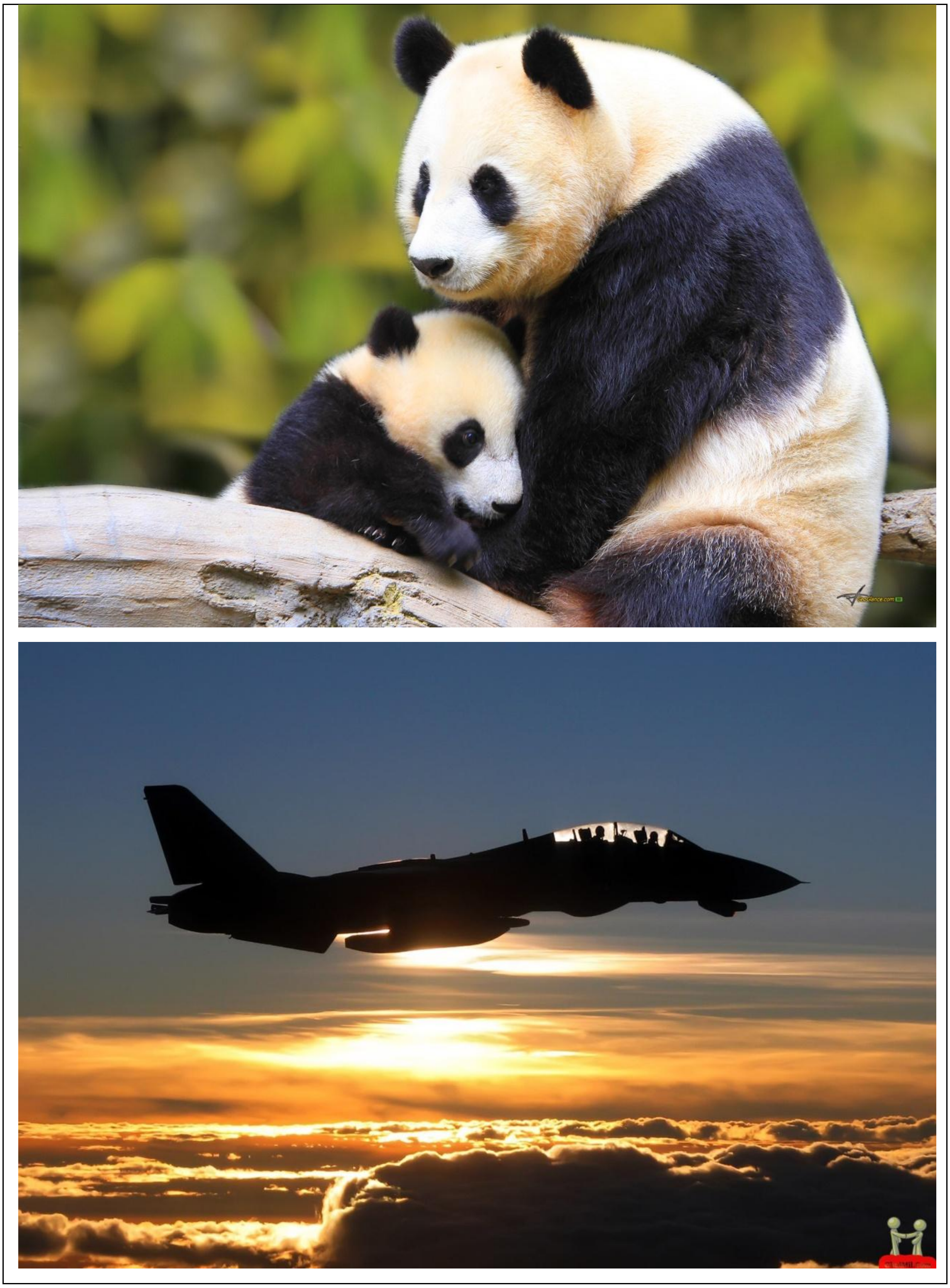


Fig. 3.4.1.2

Out of the above two images (Fig. 3.4.1.2), women are likely to prefer the first one while men are likely to prefer the second one.

3.5 Other Applications of the Game

A good global ranking can be extracted because, interestingly, this is the same problem as inferring the skill of chess players by just looking at their wins and losses. Extracting global rankings of large collections of images based on ―beauty‖ has applications to image search and computer vision. In the case of image search, knowing which images are more appealing could allow for a search engine that displays the more appealing images first. In the case of computer vision, this data could be used to train algorithms that automatically assess the quality of an image (e.g., a camera that tells the user how good a picture is).

The second could be that, after a person has played the game on a small number of pairs of images, it is possible to extract the person's general image preferences. This problem is known as collaborative filtering and is well-studied for the case of users giving absolute ratings (e.g., assigning a numerical score to each image). We present a new algorithm for collaborative filtering that needs only relative judgments between pairs of images, and we show that our algorithm is better at predicting the users' behaviour than global rankings that do not distinguish among different users. This implies that user preferences on images are, as expected, subjective.

3.5.1 GLOBAL RANKINGS

We first look at several methods to combine the relative judgments into a global ranking. For the global ranking, we consider the data as a multidigraph $G= (V, E)$- a directed graph which is permitted to have multiple arcs between two nodes. The nodes V are the images. For every decision made by a user to prefer image $i \in V$ over a different image $j \in V$ there exists a directed edge $i, j \in E$. The goal then is to produce a global ranking $\boldsymbol{r}=\boldsymbol{r1}>\boldsymbol{r2}>\boldsymbol{r3}>\cdots>\boldsymbol{rn}$ over all of the images. The following methods all have in common that they use a *ranking function* $f:V\mapsto\mathbb{R}$ that maps every image to a real number first, called its *rank value*, and then applies sorting. For this *induced ranking* $\boldsymbol{r}\ \boldsymbol{f}$ it holds that an image is ahead of a different image if its rank value is larger: $\boldsymbol{r}\ \boldsymbol{fx}>\boldsymbol{r}\ \boldsymbol{fy}\Leftrightarrow f(x)>f(y)$

We compare three different ranking functions: empirical winning rate (EWR) and ELO rating.

3.5.1.1 Empirical Winning Rate (EWR)

Perhaps the simplest form of a ranking function is to use the empirical winning rate as an estimate for an image's quality. The empirical winning rate is the number of times an image was preferred over a different image, divided by the total number of comparisons in which it was included. In graph terms, the empirical winning rate of an image is just its out degree divided by its degree: f_{EWR} (i) = (deg+ i /deg i) the empirical winning rate is easy to understand, but has two problems. For images that have a low degree (because they have taken part in few comparisons), the empirical winning rate might be artificially high or low. The second problem is that it does not take the quality of the competing image into account, i.e., winning against a bad image is worth the same as winning against a good image.

3.5.1.2 ELO Rating

The ELO rating tries to overcome the latter problem. The ELO rating system was introduced by Arpad Elo for rating chess players. In this model, each chess player's performance in a game is modelled as a normally distributed random variable. (We note that later studies showed logistic distribution to be a better model for chess rankings.) The mean of that random variable should reflect the player's true skill and is called the player's ELO rating. If a player wins, his/her ELO rating goes up, otherwise it goes down. The actual difference depends on how good the other player is, i.e., how surprising a win or loss is.

For learning, we first initialize each image's ELO rating Ri to 1,600. Before each comparison between two images i and j we compute their *expected scores* (i.e., their expected chance of winning in this comparison) E_i and E_j according to a scaled logistic function:

E_i= 1/(1+10$^{(Ri^{-}Rj/^{400})}$) E_j=1/(1+10$^{Rj^{-}Ri/^{400}}$)

The factor 400 is chosen such that a player whose ELO score is 200 higher than another player's ELO score has a chance of winning of about 75%. After the comparison, we know that either image i or image j won, i.e., we know the *true score* Si: Si=1 if image i won and Si=0 if image i lost. The prediction error is $Si−Ei$. We then update the image's ELO rating Ri accordingly: $Ri⟵Ri+K(Si−Ei)$

Thus, if the *expected score* of image i is above its *true score* the image's ELO rating will be adjusted downward, otherwise it will be adjusted upward. K is a model parameter that defines by how much the scores of the two images are changed. A large value of K makes the scores more sensitive to winning or losing a single comparison. To compute the ELO ratings, we iterate over all comparisons in our training set and update the Ri's accordingly. We then use the image's ELO ratings as our ranking function: $f_{ELO}\ i = E_{i.}$

Chapter 4 CODE

4.1 Connection to Database

```
<?php
$con=mysql_connect("localhost","root","");
if(!$con){
        die('could not connect'. mysql_error());
}
mysql_select_db("pal",$con);
?>
```

4.2 Registration

```
<?php
include 'connect1.php';
session_start();
?>

<html>
<head>
<title> Registration </title>
<style>
h2{
        font-size: 72;
        color: WHITE;
        margin-left:800px;
        font-family: AR DARLING;
        font-weight: bold;
}

h4{
        font-size: 15;
        color: white;
        text-decoration: underline;
}
h1{
        font-size: 50;
        color: WHITE;
        font-family: Algerian;
        font-weight: bold;
}
th{
        text-align: right;
        color: WHITE;
        font-size: 42;
        font-family: Agency FB;
}

body{
        background-image:url("images/lb.jpg");
}
legend{
        font-size: 35;
        color: WHITE;
        margin-left: 900px;
        font-weight: bold;
}
</style>
</head>
<body>
<h2> Matchin </h2>
<div>
<form action="reg_action.php" method="post">
<fieldset>
<legend>Register for Free!</legend>
```

```
<table cellspacing="15" style="margin-left: 220px;">
<tr>
<th> User Name: </th>
<td></td>
<td></td>
<td></td>
<td></td>
<td> <input type="text" size="50" max size="10" name="use"> </td>
</tr>
<tr>
<th> Gender: </th>
<td></td>
<td></td>
<td></td>
<td></td>
<td>
<input type="radio" name="gen" value="M">Male
<br>
<input type="radio" name="gen" value="F">Female
</td>
</tr>
<tr>
<th> Email Id: </th>
<td></td>
<td></td>
<td></td>
<td></td>
<td> <input type="text" size="50" name="ema"> </td>
</tr>
<tr>
<th> Password:</th>
<td></td>
<td></td>
<td></td>
```

```
<td></td>
<td> <h4> Atleast 8 characters required </h4> <input type="password" size="50" name="pas"> </td>
</tr>
<tr>
<th> Re enter Password: </th>
<td></td>
<td></td>
<td></td>
<td></td>
<td> <input type="password" size="50" name="rep"> </td>
</tr>
</table>
<p align="center"><input type="submit" value="Register Now"></p>
</fieldset>
</form>
</body>
</html>
```

4.3 Register-Action

```
<html>
<head><title>Register</title>
</head>
<body>
<?php
session_start();
include 'connect1.php';

$uname=$_POST['use'];
$gen=$_POST['gen'];
$email=$_POST['ema'];
$pswd=$_POST['pas'];
$repswd=$_POST['rep'];
```

```
$enc=md5($pswd);

$x=strlen($pswd);
if($uname=="" || $gen=="" || $email=="" || $pswd=="" || $repswd=="")
{
echo "Please fill in all the fields";
}

if($pswd != $repswd)
{
        echo "Passwords do not match. Please re enter Passwords";
        exit("");

}

if($x < 8)
{
        echo "Password too short";
        exit("");
}

if (!preg_match("/^([a-zA-Z0-9_\.\-])+\@(([a-zA-Z0-9\-])+\.)+([a-zA-Z0-9]{2,4})+$/",$email))
{
        echo "Please enter a valid email id";
        exit("");
 }

$result = mysql_query("SELECT username FROM registeration");
$storeArray = Array();
while ($row = mysql_fetch_array($result, MYSQL_ASSOC)) {
   $storeArray[] =  $row['username'];
}
```

```
while(next($storeArray))
{
        if(current($storeArray)==$uname)
        {
        echo "Username is taken";
        exit("");
        }
}

$query="INSERT INTO registeration (username, Gender, email, password) VALUES
('$uname', '$gen','$email','$enc')";
mysql_query($query);

echo "New record entered successfully";
?>
<script language="javascript">
window.open("home.php");
</script>
</body>
</html>
```

4.4 Login/Home Page

```
<html>
<head>
<title> Home </title>
<style>
a:link {
   color: #FF0000;
}

a:visited {
   color: #00FF00;
```

```
}
body{
        background-image:url("images/lb.jpg");
}
h2{
        font-size: 72;
        color: WHITE;
        margin-left:800px;
        font-family: AR DARLING;
        font-weight: bold;
}
th{
        text-align: right;
        color: WHITE;
        font-size: 42;
        font-family: Agency FB;
        font-weight: bold;
}
legend{
        font-size: 35;
        color: WHITE;
        margin-left: 900px;
        font-weight: bold;
}
p{
        color: WHITE;
        font-size: 20;
        margin-left: 220px;
        font-weight: bold;
}
</style>
<script>
function check()
{
```

```
window.open("start.php");
}
</script>
</head>
<body>
<h2>Matchin</h2>
<form action="login.php" method="post">
<fieldset>
<legend> Log In </legend>
<table style="margin-left: 220px;">
<tr>
<th> User Name: </th>
<td></td>
<td></td>
<td></td>
<td></td>
<td> <input type="text" size="50" name="user"> </td>
</tr>
<tr>
<th> Password: </th>
<td></td>
<td></td>
<td></td>
<td></td>
<td> <input type="password" size="50" name="pass"> </td>
</tr>
</table>
<p align="center"><input type="submit" value="Log In"></p>
</fieldset>
</form>
<p>Not a Registered User?
<a href="register.php">Register Now to Play</a>
</p>
</body>
```

```
</html>
```

4.5 Login-Action

```
<html>
<head>
</head>
<body>
<?php

session_start();
include 'connect1.php';

$uname=$_POST['user'];
$pswd=$_POST['pass'];
$f=1;

$enc=md5($pswd);

$query="SELECT * FROM registeration WHERE username='$uname'";
$result=mysql_query($query);
$numrows = mysql_num_rows($result);
if ($numrows!=0)
{
while ($row=mysql_fetch_assoc($result))
{
        $dbuname = $row['username'];
        $dbpswd = $row['password'];
}

if($uname==$dbuname && $enc==$dbpswd)
{
        $t=time();
        $_SESSION['username']=$uname;
        $_SESSION['flag']=$f;
        $_SESSION['logintime']=$t;

        $filename="talktod".$uname.".php";
        fopen($filename, "w");
        copy("talktodb1.php",$filename);
        include $filename;

        $file="start".$uname.".php";
        fopen($file, "w");
        copy("start.php",$file);
        include $file;
}

else
{
echo "Incorrect password!";
exit();
}
}
else{
echo "The user doesn't exist!";
exit();
}
?>
</body>
</html>
```

4.6 Start Page

```
<?php
include 'connect1.php';

        $t=$_SESSION['logintime'];
```

```
            $uname=$_SESSION['username'];
            $f=$_SESSION['flag'];

            $filename="match1".$uname.".php";
            fopen($filename, "w");
            copy("match.php",$filename);

            $statement=$filename."?un=".$uname." & fl=".$f." & lt=".$t;

?>

<html>
<head>
<title> Start </title>
<link rel="stylesheet" type="text/css" href="mystyle/basic.css">
<style>
body{
            background-image:url("images/lb.jpg");
}

p{
            font-size:65;
            font-family:AR DARLING;
            font-weight: bold;
            color: black;
}
div {
    -moz-appearance:button; /* Firefox */
    -webkit-appearance:button; /* Safari and Chrome */
    appearance:button;
            height: 200px;
            width: 500px;
            margin-left: 440px;
            margin-top:90px;
}
</style>
</head>
<body>
<h2> Matchin </h2>
<a href="<?php echo $statement ?>"><div>
<p align="center">Start Playing!</p>
</div>
</body>
</html>
```

4.7 Match Players

```
<html>
<head>
<title> Start </title>
<link rel="stylesheet" type="text/css" href="mystyle/basic.css">
<style>
body{
        background-image:url("images/lb.jpg");
}

p{
        font-size:65;
        font-family:AR DARLING;
        font-weight: bold;
        color: black;
}
</style>
</head>
<body>
<!--<p align="center"><?php echo "Waiting to Match..."; ?> </p>-->
</body>
</html>
```

```
<?php

include 'connect1.php';

$t=time();
$_SESSION['logintime']=$t;



$uname=$_GET['un'];
$ff=$_GET['fl'];
$tt=$t;

echo $uname;

$qup="UPDATE registeration SET lastlogin='$t' WHERE username='$uname'";
mysql_query($qup);

include 'matched.php';

if($n>1)
{
        sort($ll);
                /*match $ll[0] and $ll[1] and so on...*/

        if($n%2!=0)
        $n=$n-1;

        for($i=0;$i<$n;$i=$i+2)
        {
                if($gid[$i]==NULL)
                {
                $j=$i+1;
                $quer="UPDATE registeration SET game_id='$i', ugid='2' WHERE lastlogin='$ll[$i]'";
                mysql_query($quer);
                $quer1="UPDATE registeration SET game_id='$i', ugid='1' WHERE lastlogin='$ll[$j]'";
                mysql_query($quer1);
                }
        }
        $_SESSION['username']=$uname;
        $_SESSION['gid']=$i;

        $filegame="random".$uname.".php";
        fopen($filegame, "w");

        $pq="SELECT ugid FROM registeration WHERE username='$uname'";
        $rs=mysql_query($pq);
        $r=mysql_fetch_row($rs);

        if($r[0]==1)
        copy("random.php",$filegame);

        if($r[0]==2)
        {
        $pname=$uname;
        copy("random2.php",$filegame);
        }

        $qu="SELECT game_id FROM registeration WHERE username='$uname'";
        $re=mysql_query($qu);
        $da=mysql_fetch_row($re);

        echo $da[0];
        if($da[0]!=NULL)
        {
        include $filegame;
        }
```

```
        else
        echo "No Match!";
}

else
echo "no match";
?>
```

Matched.php

```
<?php

include 'connect1.php';

$n=0;

$query="SELECT id, username, lastlogin, game_id FROM registeration WHERE flag='1' ";
$result=mysql_query($query);
while($data=mysql_fetch_row($result))
{       $id[$n]=$data[0];
        $name[$n]=$data[1];
        $ll[$n]=$data[2];
        $gid[$n]=$data[3];
        $n++;
}
?>
```

4.8 Select random pairs of images

Player 1-

```
<?php
session_start();
include 'connect1.php';
$uname=$_SESSION['username'];
$path="imagescomp/";
```

```
$result=mysql_query("SELECT * FROM photos WHERE 1");
$n=mysql_num_rows($result);

for($k=0;$k<10;$k++)
{
        $z=0;
        $c1=mt_rand(1,$n);
        $c2=mt_rand(1,$n);
        if($c1==$c2)
        {
        while($c2==$c1)
        $c2=mt_rand(1,$n);
        }
        if($k==0)
        {
        $arr[$k][0]=$c1;
        $arr[$k][1]=$c2;
        }
        if($k>0)
        {
                for($m=0;$m<$k;$m++)
                {

        if(($arr[$m][0]==$c1&&$arr[$m][1]==$c2)||($arr[$m][0]==$c2&&$arr[$m][
1]==$c1))
                        {
                                $z=1;
                                $k=$k-1;
                                break;
                        }
                }

                if($z==0)
                {
```

```
            $arr[$k][0]=$c1;
            $arr[$k][1]=$c2;
        }
    }

}
$_SESSION['ran']=$arr;
$jj=0;
$_SESSION['counter']=$jj;
for($ii=0;$ii<10;$ii++)
{
    $fin="display1".$uname.$ii.".php";
    fopen($fin, "w");
    copy("display1.php",$fin);
}
$ffin="display1".$uname."0".".php";
include $ffin;
?>
```

Player 2-

```
<?php
session_start();
$random=$_SESSION['ran'];
$jj=$_SESSION['counter'];
for($ii=0;$ii<10;$ii++)
{
    $fin="display1".$uname.$ii.".php";
    fopen($fin, "w");
    copy("display1.php",$fin);
}
echo "name in random 2- ";
echo $pname;
echo "<br>";
echo $uname;
$ffin="display1".$uname."0".".php";
```

```
include $ffin;
?>
```

4.9 Game Page

```
<?php
$img=$_SESSION['ran'];
$cc=$_SESSION['counter'];

$cc2=$cc;
$t1=time();

$ci1=$img[$cc2][0];
$ci2=$img[$cc2][1];

$r1=mysql_query("SELECT id FROM photos WHERE sno='$ci1'");
$r2=mysql_query("SELECT id FROM photos WHERE sno='$ci2'");

$im1=mysql_fetch_row($r1);
$im2=mysql_fetch_row($r2);

$path=$path="imagescomp/";

$sub="gotodb".$uname.".php";
fopen($sub, "w");
copy("gotodb.php",$sub);

echo "Name in display is- ";
echo $uname;

?>
<html>
<head>
<title> GAME </title>
```

```
<style>
body{
	background-image:url("images/lb.jpg");
}
h2{
	font-size: 72;
	color: WHITE;
	margin-left:1000px;
	font-family: AR DARLING;
	font-weight: bold;
}
th{
	text-align: right;
	color: WHITE;
	font-size: 42;
	font-family: Agency FB;
	font-weight: bold;
}
p{
	color: WHITE;
	font-size: 20;
	margin-left: 150px;
	font-weight: bold;
}
h1{
	color: WHITE;
	font-size: 40;
	margin-left: 170px;
	font-weight: bold;
}
</style>
</head>
<body>
<h2>Matchin</h2>
```

```
<h1>Which image do you think your partner will prefer?</h1>
<p><img src="<?php echo $path.$im1[0]; ?>" width="400" height="400">
<img src="<?php echo $path.$im2[0]; ?>" width="400" height="400" style="margin-left:130px;"></p>
<form action="<?php echo $sub ?>" method="post">
<input type="hidden" name="player" value="<?php echo $uname ?>">
<input type="hidden" name="time1" value="<?php echo $t1 ?>">
<input type="hidden" name="first" value="<?php echo $im1[0] ?>">
<input type="hidden" name="second" value="<?php echo $im2[0] ?>">
<input type="hidden" name="counterval" value="<?php echo $cc2?>">
<input type="submit" name="same" value="This One!" style="margin-left:300px">
<input type="submit" name="same" value="That One!" style="margin-left:480px">
</form>
</body>
</html>
```

4.10 Saving Results to Database (gotodb.php)

```
<?php
session_start();
include 'connect1.php';

$t2=time();
$t1=$_POST['time1'];
$tw=$t2-$t1;

$iname=$_POST['same'];
$playername=$_POST['player'];
$i1=$_POST['first'];
$i2=$_POST['second'];
$tnow=time();

if($iname=="This One!")
{
        $bi=$i1;
        $wi=$i2;
}

else
{
        $bi=$i2;
        $wi=$i1;
}
$qnew="SELECT game_id FROM registeration WHERE username='$playername'";
$diff=mysql_query($qnew) or die (mysql_error());

$gameid=mysql_fetch_row($diff);

$qnew1="SELECT id FROM registeration WHERE username='$playername'";
$diff1=mysql_query($qnew1);
$pid=mysql_fetch_row($diff1);

$inres="INSERT INTO results (time, gid, userid, better, worse, wait) VALUES
('$tnow','$gameid[0]','$pid[0]', '$bi', '$wi', '$tw')";
mysql_query($inres);

$ccf=$_POST['counterval'];

if($ccf<9)
{
$ccf++;
$_SESSION['counter']=$ccf;
$next="display1".$playername.$ccf.".php";
$uname=$playername;
include $next;
}
if($ccf==9)
{
```

```
$score="score".$playername.".php";
fopen($score, "w");
copy("score.php",$score);
include $score;
}
?>
```

4.11 Check Database

```
<html>
<head><title>checkdb</title>
<head>
<body>
<table border="2" cellspacing="10">
<tr>
<th>Id</th>
<th>Username</th>
<th>Gender</th>
<th>Email</th>
<th>Password</th>
<th>Flag</th>
<th>Game_id</th>
<th>LastLogin</th>
<th>Player_id</th>
</tr>
<?php
session_start();
include 'connect1.php';

$query="SELECT * FROM registeration WHERE 1";
$result=mysql_query($query);
while($data=mysql_fetch_row($result))
{
        echo("<tr><td>$data[0]</td><td>$data[1]</td><td>$data[2]</td><td>$data[3
]</td><td>$data[4]</td><td>$data[5]</td><td>$data[6]</td><td>$data[7]</td><td>$
data[8]</td></tr>");
}
?>
</table>
<br> <br>
<table border="2" cellspacing="10">
<tr>
<th>photo id</th>
</tr>
<?php
include 'connect1.php';

$quer="SELECT * FROM photos WHERE 1";
$resul=mysql_query($quer);
while($data=mysql_fetch_row($resul))
{
        echo("<tr><td>$data[0]</td><td>$data[1]</td></tr>");
}

?>
</table>
<br> <br>
<h2>Result Set</h2>
<table border="2" cellspacing="10">
<tr>
<th>decision_id</th>
<th>time</th>
<th>gid</th>
<th>userid</th>
<th>better</th>
<th>worse</th>
<th>wait time</th>
```

```
</tr>
<?php
include 'connect1.php';

$que="SELECT * FROM results WHERE 1";
$resu=mysql_query($que);
while($dat=mysql_fetch_row($resu))
{
echo("<tr><td>$dat[0]</td><td>$dat[1]</td><td>$dat[2]</td><td>$dat[3]</td><td>
$dat[4]</td><td>$dat[5]</td><td>$dat[6]</td></tr>");
}

?>
</table>
</body>
</html>
```

4.12 Image Database:-

4.12.1 Image Insertion Form

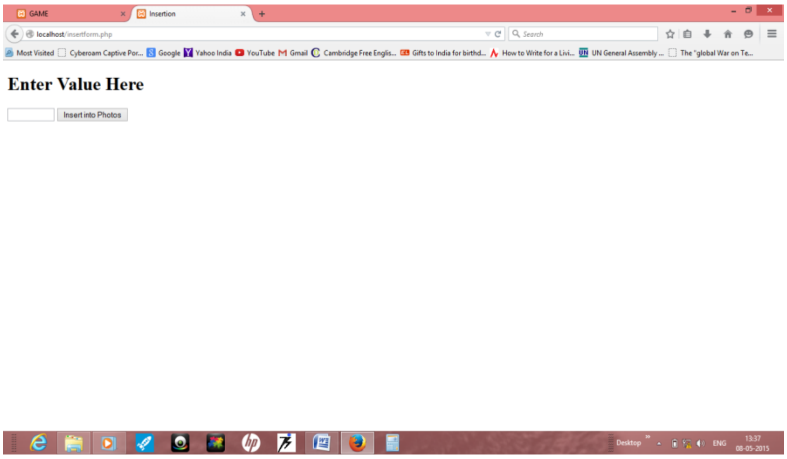
```
<html>
<head>
<title> Insertion </title>
</head>
<body>
<form action="insert.php" method="post">
<h1>Enter Value Here</h1>

<input type="text" size="10" max size="10" name="pid">
<input type="submit" value="Insert into Photos">

</form>

</body>
```

```
</html>
```



Enter Value Here

Fig. 4.8.1.1

4.12.2 Image Insertion.php

```
<html>
<head>
</head>
<body>
<?php
include 'connect1.php';

$id=$_POST['pid'];
$path="C:\\\\xampp\\\htdocs\\\imagescomp";

$query="INSERT INTO photos (id) VALUES ('$id')";
mysql_query($query);

echo "New record entered successfully";
?>
<body>
</html>
```

4.13 Destroy Session Variables after Game is over

```
<?php
include 'connect1.php';
$query="SELECT username FROM registeration WHERE flag='1'";
$result=mysql_query($query);
$data=mysql_fetch_row($result);

$quer="SELECT lastlogin FROM registeration WHERE flag='1'";
$res=mysql_query($quer);
$dat=mysql_fetch_row($res);
$ss=session_status();
if($ss==1)
{
        $q="UPDATE registeration SET flag='0' WHERE username='$data[0]'";
                mysql_query($q);
}
?>
```

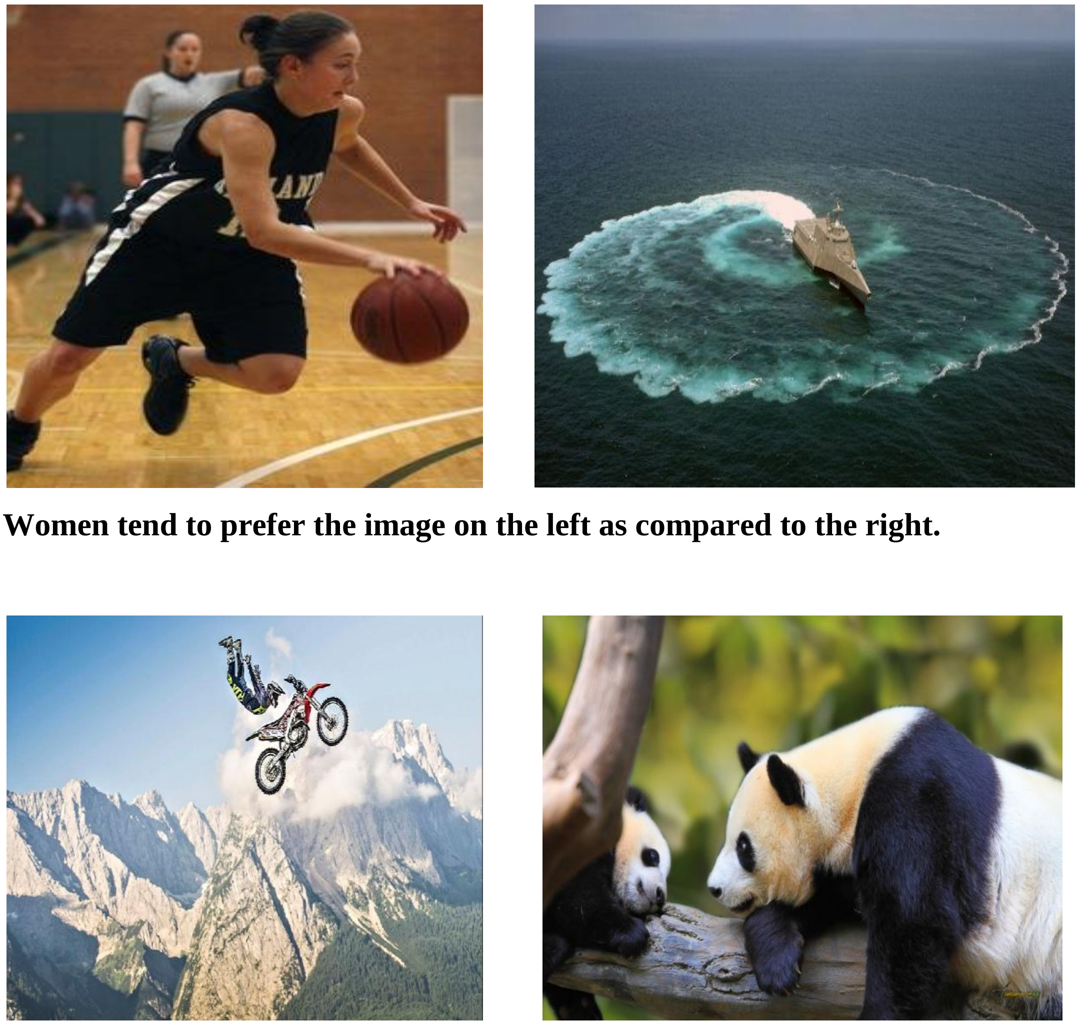
4.14 Tracking User in Database (talktodb1.php)

```
<?php
include 'connect1.php';

$uname=$_SESSION['username'];
$f=$_SESSION['flag'];
$t=$_SESSION['logintime'];
$copy=$uname;

$q="UPDATE registeration SET flag='$f', lastlogin='$t' WHERE username='$uname'";
mysql_query($q);
include 'runalways1.php';
?>
```

RESULTS AND GENERAL TRENDS



Women tend to prefer the image on the left as compared to the right.

Men tend to prefer the image on the left as compared to the image on the right.

Fig. 5.1

Generally, the pictures that made it into the top 100 are neither provocative nor offensive. This could mean that since the players do not know their partner (or their gender) they go for a ―safe choice. Most of the highly ranked pictures express peaceful and harmonious environs (like a waterfall or a sunset). This suggests that people think a random person will most likely prefer peace and harmony.

CONCLUSION & FUTURE WORK

The main contribution of this project is to provide a method to elicit user preferences. For two images, we ask users not to tell which one they prefer, but rather which one a random person will prefer. We reward them if they are correct in their prediction. We compared several algorithms for combining these relative judgments into a total ordering.

Finally, we present a gender test that asks users to make some relative judgments and, based only on these judgments, we can predict a random user’s gender.

It remains to be investigated how much other personal information can be gathered in the same way as our gender test does.

REFERENCES

1. Luis Von Ahn, “Games With A Purpose”, IEEE Computer Magazine, June 2006, pp 96-98
2. Laura Dabbish, Luis Von Ahn, “General Techniques for Designing Games with a Purpose”, Communications of the ACM, August 2008. pp 58-67.
3. Luis Von Ahn, “Eliciting User Preferences through an Online Game”, ACM Conf. on Human Factors in Computing Systems, CHI 2009. pp 1207-1216
4. Alexander J. Quinn, Benjamin B. Bederson, “Human Computation: A Survey and Taxonomy of a Growing Field”, Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI 2011. pp 1403-1412
5. www.w3schools.com
6. http://en.wikipedia.org/wiki/Luis_von_Ahn