An Interactive Framework for Image Annotation through Gaming

Lasantha Seneviratne
Multimedia and Vision Research Group, School of Electronic Engineering and Computer Science, Queen Mary University of London
Mile End Rd, London
E1 4NS, UK
Lasantha.s@elec.qmul.ac.uk

Ebroul Izquierdo
Multimedia and Vision Research Group, School of Electronic Engineering and Computer Science, Queen Mary University of London
Mile End Rd, London
E1 4NS, UK
ebroul.izquierdo@elec.qmul.ac.uk

ABSTRACT

Image indexing is one of the most difficult challenges facing the computer vision community. Addressing this issue, we designed an innovative approach to obtain an accurate label for images by taking into account the social aspects of human-based computation. The proposed approach is highly discriminative in comparison to an ordinary content-based image retrieval (CBIR) paradigm. It aims at what millions of individual gamers are enthusiastic to do, to enjoy themselves within a social competitive environment. It is achieved by setting the focus of the system on the social aspects of the gaming environment, which involves a widely distributed network of human players. Furthermore, this framework integrates a number of different algorithms that are commonly found in image processing and game theoretic approaches to obtain an accurate label. As a result, the framework is able to assign (or derive) accurate tags for images by eliminating annotations made by a less-rational (cheater) player. The performance analysis of this framework has been evaluated with a group of 10 game players. The result shows that the proposed approach is capable of obtaining a good annotation through a small number of game players.

Categories and Subject Descriptors

H.5.3 [**Group and Organization Interfaces**]: Computer-supported cooperative work; I.2.1 [**Application and Expert Systems**]: Games

General Terms

Algorithms and Design

Keywords

Semantic annotation, Interactive gaming, Human computation, MPEG-7 features and object recognition.

MIR'10, March 29–31, 2010, Philadelphia, Pennsylvania, USA.

1. INTRODUCTION

Recent developments in social networks have contributed to the already large quantity of digital multimedia content on the World Wide Web (WWW). As a consequence, the following questions arise, do people label the content? If so, how often do they do so? Reacting to these and other similar questions, researchers around the world have designed a considerable number of algorithms and frameworks with the capabilities of automated image tagging [1]. However, due to the semantic gap, finding a general solution for image tagging has become a challenge. Over the last decade, a number of research directions have been explored addressing the semantic gap problem. One such approach is to divide a complex task that cannot be achieved by an algorithm into a reasonable number of chunks. From this point, collaborative effort can be exploited for the purpose of image tagging [2] [3] [4].

Since the ESP game [5] was introduced, a number of similar approaches to address the semantic gap issue have been proposed. Including the ESP game, most of the other approaches use humans in image tagging. Among them, the ASAA (Application for Semi-Automatic Annotation) [6] and "Manhattan Story Mashup" [2] are two different game strategies introduced in the literature. These strategies have extended the collaborative work paradigm into another era by introducing two different methods of harvesting human brainpower. The first approach to engaging human attention is designing interactive frameworks with multiplayer game strategies. It has been shown to be fun and entertaining. As a result, public attention is drawn into playing the game and its real purpose, image annotation, goes largely unnoticed. However, when it comes into practice, multiplayer game strategies present their own challenges. For example, the ESP game is used to annotate images using two similar key words given by two unseen players. This approach is a highly effective if players do not cheat by entering unrelated keywords such as "cat" for every image [7], leading the system to generate no useful information.

While focusing on benefits of a collaborative work, we introduced a novel approach, a standalone game capable of fulfilling the major objective in computer vision community, the semantic tagging of images. This approach is also capable of entertaining a game player thus it can reduce the cost of manual annotation. As a result, the framework obtains good annotations based on human perception of images. In this paper, we describe a game designed to implement this framework. It is a standalone game that enhances accuracy in image annotation by using humans in the loop and also makes use of cheat-prevention techniques. The system identifies less-rational game players, by taking in to account the previous outcomes. We proposed two approaches for player outcome. The first one used a Markov Model (MM) [8] [9] and second approach used the Bayes theorem [10] to measure the probabilistic distribution of accuracy of a player outcome. Additionally, the MM was used to measure the player contribution and cost before calculating their payoffs. Using these models, the framework decides the most suitable image to expose a player (loading of a fully annotated or a non-annotated image). In fact, this prediction model increased the performance of the game by predicting a player's outcome. Furthermore, we introduce a game theoretical approach to accept or to reject a player keyword by using the payoff functions. This approach analyses a player outcome by assuming the proposed framework is a two-player (virtual) game model. Therefore, the payoff functions of the game have been calculated by allowing the 'real' game player to take the role of player 1 and the framework to take the role of player 2. Then we create a payoff profile by taking into account the player contributions and their costs. This payoff profile investigates the Nash Equilibrium [22] in each game round and it decides whether to accept or to reject a player outcome (annotation). Finally, this game is designed to reward players for their contribution, granting them some game points in order to encourage them towards more gaming.

The rest of this paper is organized as follows. Section 2 presents a brief summary of the related work. Section 3 introduces the two-player virtual game model and analysis of equilibrium. In section 4, the evaluation of gaming environment based on a two-player game model is discussed. Finally section 5 summarizes the paper along with the future research goals.

2. RELATED WORK

Recent research in the computer vision community has focused on designing attractive games to capture the human contribution towards image annotation. This idea has been well proven by the statistical results of the ESP game [5]. The ESP game is a multiplayer network game, designed to be played by two game players selected on a random basis. The players are shown the same image and requested to comment on it. When they have agreed on a keyword, the image will be temporally annotated. Furthermore, this process will be continued until an image has been described by a reasonable number of game players. The ESP can be listed as the first game designed for image annotation. Since then, a number of similar approaches have been introduced for example, ASAA (Application for Semi-Automatic Annotation) [6], Manhattan Story Mashup [2], Peekaboom [11], KissKissBan [12], Verbosity [13], Phetch [14], HOT or NOT [15] and Google Image Labeller [16]. These games are designed to annotate images while providing entertainment via interaction. Game players won't realize that they are making a contribution to annotation, but they do play games to entertain themselves. As shown in Figure 1, these frameworks can be divided into a number of different groups such as image annotation by objects, locations, scene wise or player likeness.

There are several major differences to distinguish our approach from other related work. It is an approach mainly based on a standalone game fulfilling the following requirements: a game that people can use when they have idle time to spend (such as waiting in airports, train stations, hospitals, etc), it should be entertaining and interactive for the majority of game players, the efficiency of the game in image annotation should be a reasonable figure (we do not want just to entertain players).

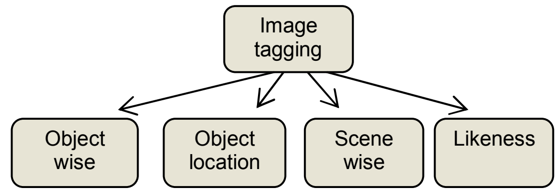


Figure 1. Categorization of image tagging approaches.

The main objective of our research is to implement an efficient framework (a game) to harness human brainpower towards the image annotation. We believe labels obtained by our framework will be used in a variety of applications, such as in image search engines, machine learning applications, educational and broadcasting purposes, etc.

3. OUR APPROACH

We introduce our framework, a computer game that is designed to harness human brainpower towards image annotation. To obtain an accurate annotation, this framework combines a number of key paradigms such as image processing, machine learning and game theoretic approaches. The issues that we address in order to fulfill the objectives of our work can be divided into a number of categories: creating a visual game interface (a game scenario capable of entertaining game players); implementing a framework to predict a player outcome (thereby the framework could discard any annotations made by a less-rational game player); designing a framework to accept or to reject an annotation made by a player (this process could filter out a keyword made by a player by taking into account of their contributions and other related factors); optimizing the framework to achieving a good annotation. During the implementation phase, we have seriously taken these factors into consideration to implement a successful framework for image annotation. As shown in Figure 2, there are players with different attitudes such as the players who provide bad tags that our framework is capable of detecting.

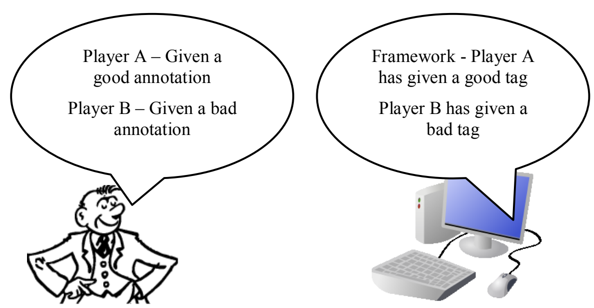


Figure 2. A player dynamics in a real game scenario.

In practice, players are used to improve their payoffs by providing good annotations. However, some players will try to improve their payoffs believing that cheating can improve their utilities. Therefore, cheat preventing can be described as a fundamental problem that has to be address while designing a framework for image annotation.

We have introduced an interactive framework for image annotation that fulfils the above mentioned factors. Figure 3 illustrates the complete framework of the proposed approach. The visual interface displays the image subject for annotation and players are intended to comment on it using a keyword.

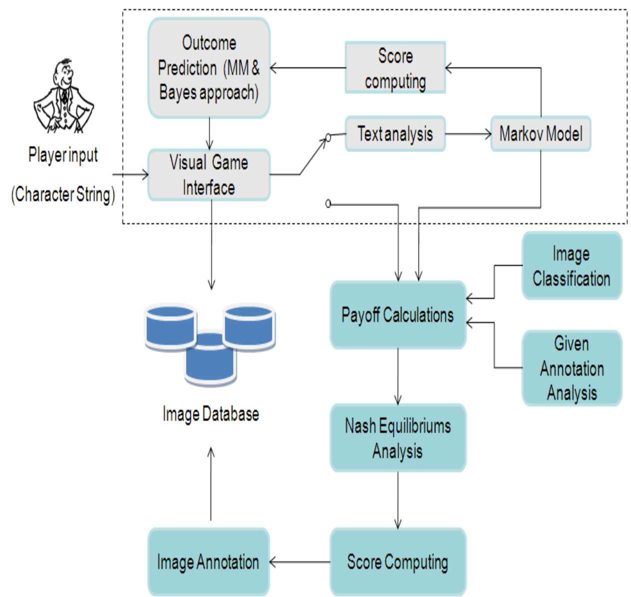


Figure 3. A complete block diagram of the framework.

As shown in Figure 3, the complete framework has been comprised of two modules. The first module (top) analyses the annotations made for fully-annotated images and the bottom section analyses the annotations made for non-annotated images.

The player prediction (outcome prediction) is another important module which predicts a player outcome by taking into account of previous annotations. The payoff calculation section aggregates all other relevant information such as player confidence, image classification and previous annotations to obtain a good annotation. The Nash analysis module decides whether to accept or to reject a player annotation by analyzing the Nash equilibriums of the players. As a way to reward a player, the score computation module calculates the player score by taking into account their contribution in image annotation.

3.1 Visual Interface

The visual interface is an important component of a game. In fact, it entertains and encourages a player into gaming. Considering these factors, we have designed a simple and an effective game scenario to involve players in gaming. Furthermore, to find the contribution of a wide audience, we have proposed two different game approaches. One approach is a less interactive scenario, where players have to comment on images by typing keywords. The other approach can be categorized as a highly interactive approach. Here, a player has to enter keywords by collecting each single character from a series of dropping characters. To improve the game efficiency, the visual interface displays a number of spinning characters. These characters fall from the top to bottom of the screen. To collect characters, a player has to use the keyboard. As an example, if the player wants to create the word "CAT", he has to collect each character "C","A" and "T" in a sequential order. To improve the game efficiency, visual interface displays 4 to 5 characters at a time. Furthermore, it is used to generate magic characters, which are changeable into any character as the player demanded. To fulfill a player requirement, the system allows a player to change the speed of spinning characters. However, if the player couldn't enter a keyword in a given time period, he will be asked to restart the game.

To maintain a list of Top-Ten game players, the players have been asked to register themselves via a username and a password. This strategy allows the framework to keep a record of metadata related to a player contribution. In Figure 4, a set of game screen shots are presented.

Figure 4. Game interface.

3.2 Player Outcome Prediction

As previously introduced, two different player prediction approaches were implemented (one using a MM and the other by Bayes theorem). These two approaches were used in the framework to decide the most suitable image to the player (fully annotated or non-annotated). To predict a player outcome, in the beginning of the game, the proposed framework loads a series of fully annotated images for the player. Additionally, it loads a number of fully annotated images to the player in random time intervals. The feedbacks from these images were used by the framework to create a MM, where it represents the player behavior. This MM is used in the framework for two purposes; firstly to predict a player outcome; secondly to calculate the payoff functions of the two-player game model. Furthermore, this MM gives following information about a player.

- $P(R|R)$ - Probability of having a "right" annotation, when previous annotation is "right".
- $P(R|W)$ - Probability of having a "right" annotation, when previous annotation is "wrong".
- $P(W|W)$ - Probability of having a "wrong" annotation, when previous annotation is "wrong".
- $P(W|R)$ - Probability of having a "wrong" annotation, when previous annotation is "right".

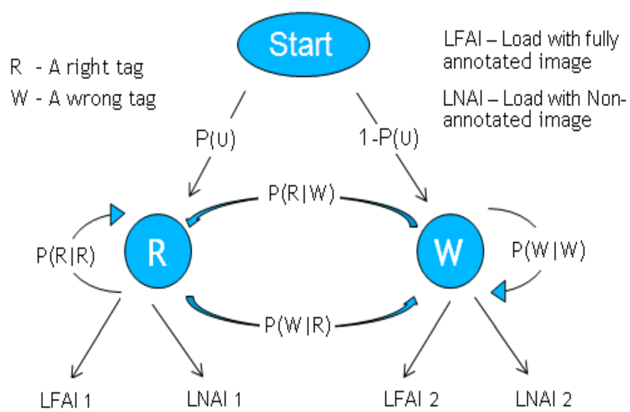


Figure 5. MM for player outcome prediction.

Figure 5 shows the MM which was used in the framework for outcome prediction. The probability $P(U)$ = Player confidence (number of "right" annotations made by a player / number of fully annotated images given to the player).

3.2.1 Player outcome prediction by Markov Model

This part of the framework predicts a player outcome using the MM. The idea behind this strategy is to construct a transition matrix regarding a player outcome for a series of fully annotated images. This was done by assuming the players respond to a non-annotated image in the same way as they do for a fully annotated image (players do not know of the image that they have been given for annotation). Furthermore, this transition matrix is used by the framework to obtain the emission probabilities of the MM. These emission probabilities were used by the framework to select the image (fully annotated or non-annotated) that is going to be loaded to the player. The process of calculating emission probabilities in this game is illustrated in eq. 1- 4.

$$LNAI\ 1 = P(R|R)P(U) \quad (1)$$

$$LFAI\ 1 = P(W|R)P(U) \quad (2)$$

$$LFAI\ 2 = P(W|W)(1 - P(U)) \quad (3)$$

$$LNAI\ 2 = P(R|W)(1 - P(U)) \quad (4)$$

An example of predicting a player outcome is given below. Let's assume the player has given a 'right' annotation. Then the probability of having a good annotation and a bad annotation can be calculated by eq. 1 and 2. If outcome of eq. 1 > eq. 2, the MM assumes the next move of the player will be a good annotation and it tells the framework to load a non-annotated image for the player in the next game round.

3.2.2 Player outcome prediction by Bayes Theorem

The second approach of predicting a player outcome was implemented using the Bayes theorem. It is presented in [10] as a model to predict an annotation, giving a set of annotations already assigned to an image. Using this approach, the probability we are interested can be written as in eq. 5. Here a 'tag set' means two sequential outcomes from a player for 2 fully annotated images ('R' means a good annotation and 'W' means a bad annotation).

In practice, players do not know of the image that they have been given for labeling. Therefore, we assume that players respond to a non-annotated image in the same way as they do for a fully annotated image. Taking this as an advantage, the Bayes approach predicts a player outcome taking into account only previous outcomes. Eq. 5 demonstrates an example of calculating the Bayes probability of having a 'right' annotation, when a 'wrong' annotation is given by the player.

$$P(\text{"}R\ is\ next\ outcome\text{"}|\text{"}W\ is\ already\ present\text{"})$$
$$= \{P(\text{"}W\ is\ already\ present\text{"}|\text{"}R\ is\ next\ outcome\text{"}) * P(\text{"}R\ is\ next\ outcome\text{"})\} / P(\text{"}W\ is\ already\ present\text{"}) \quad (5)$$

We estimate probabilities $P(\text{"}W\ is\ already\ present\text{"})$, $P(\text{"}W\ is\ already\ present\text{"}|\text{"}R\ is\ next\ outcome\text{"})$ and $P(\text{"}R\ is\ next\ outcome\text{"})$ as given in [10].

To avoid zero probability estimation, authors in [10] suggested a smooth variant p instead of using probability$P$.

$$p(\text{"}W\ is\ already\ present\text{"}|\text{"}R\ is\ next\ outcome\text{"})$$
$$= \{(1-\lambda)P(\text{"}W\ is\ already\ present\text{"}|\text{"}R\ is\ next\ outcome\text{"}) + \lambda\, P(\text{"}W\ is\ already\ present\text{"})\} \quad (5.1)$$

In all of our experiments, we used $\lambda = 0.36$, chosen using a validation dataset.

Finally,

$$P(\text{"}R\ is\ next\ outcome\text{"}|\text{"}W\ is\ already\ present\text{"})$$
$$= \{p(\text{"}W\ is\ already\ present\text{"}|\text{"}R\ is\ next\ outcome\text{"}) * (\text{"}R\ is\ next\ outcome\text{"})\} / P(\text{"}W\ is\ already\ present\text{"}) \quad (6)$$

We obtain eq. 6 as the probabilistic model for predicting a player outcome (probability of having a "right" annotation when a 'wrong' annotation is given by the player). The proposed framework used the same technique to calculate other probabilities P(R|R), P(W|R), P(R|W) and P(W|W) to predict a player outcome by taking into account the previous annotations.

3.3 Two-Player Game Model and Payoff calculation

To obtain an accurate annotation, the proposed framework predicts a player outcome. Hence, it calculates the most likely state of the observation that a player could attempt to do in a game round. However, even with the player prediction framework, still there is a probability that a player outcome could be bad. To tackle this problem, and to exploit contributions from game theory, we assumed this game represent a two player (virtual) game model. This multi-player game strategy is used in [17] as a technique to analyze a player output by taking into account the other player's outcome. We used this technique to design the framework as a multiplayer game model, assuming the framework itself is a player (player 2). Using this technique, we implemented the payoff functions of the game as follows.

Player payoff (player 1)

$$\pi_1(a_1, a_2) = a_2\ Player\ 1\ contribution - a_1 Player\ 1\ cost \quad (7)$$

Framework payoff (player 2)

$$\pi_2(a_1, a_2) = a_1\ Player\ 2\ contribution - a_2 Player\ 2\ cost \quad (8)$$

Eq. 7 and 8 defined the payoff functions of player 1 and 2. Both payoff functions consisted of two terms. The first term, i.e., $a_2\ Player\ 1\ contribution$ and $a_1\ Player\ 2\ contribution$ denotes the gain of a player with respect to the other player's action. The second term $a_1 Player\ 1\ cost$ and $a_2 Player\ 2\ cost$ demonstrates the cost of the players, in respect to their actions. The framework uses these payoff functions to analyze the labels made by a player to accept or reject it in each game round. Considering payoff functions, a_1 and a_2 represents a player action. The a_1= 1 indicates that the player 1 is willing to contribute to the game by a good annotation, $a_1 = 0$ indicates player 1 would do a bad annotation in the next game round (this action information was given by the player output prediction section). When the player 1 score is less than the threshold score (*Player score* < *Threshold score*) the action a_2 will be 0. $a_2 = 1$ indicates that the player score is greater than the threshold score (*Player score* > *Threshold score*). Here, the MM was used by the framework to calculate the player 1's contribution and cost as follows,

$$Contribution = \Big(P(R|R)P(U) + P(R|W)\big(1 - P(U)\big)\Big) \quad (9)$$

$$Cost = \Big(P(W|R)P(U) + P(W|W)\big(1 - P(U)\big)\Big) \quad (10)$$

Given this information and action profile (a_1, a_2), player payoffs are calculated as follows. Eq. 11 shows the payoff of player 1 and eq. 12 shows the payoff of player 2, calculated based on the player 1's contributions to the game.

$$\pi_1(a_1, a_2) = a_2\ \Big(P(R|R)P(U) + P(R|W)\big(1 - P(U)\big)\Big) - a_1\ \Big(P(W|R)P(U) + P(W|W)\big(1 - P(U)\big)\Big) \quad (11)$$

$$\pi_2(a_1, a_2) = a_1\ \big(\pi_1(a_1, a_2)\ + P(C) +\ P(A)\big)/I - a_2\ (NA * \ 0.083) \quad (12)$$

Here, $P(U)$ denotes the confidence level of a player as a probability, $P(A)$ represent the average probability of a given annotation, which is calculated based on other player feedbacks. The *cost* demonstrates the cost of player 1, NA denotes the number of annotations given for an image and *I* represent the normalizing constant. The *cost* for an annotation is defined by a constant of 0.083 (this is assigned to limit the maximum number of annotations per image to 12) and $P(C)$ demonstrates the probability of the outcome of image classification.

This framework used a support vector machine (SVM) [21] classifier for image classification. Furthermore, it uses Colour Layout (CLD) [18], Dominant Colour (DCD) and Edge Histogram (EHD) [19] descriptors in a linear fusion way [20] to form a high dimensional feature vector for classification.

3.4 Nash Equilibrium Representation

The Nash Equilibrium is a solution concept of a multiplayer game scenario, where each player is assumed to know the equilibrium strategies of the other player.

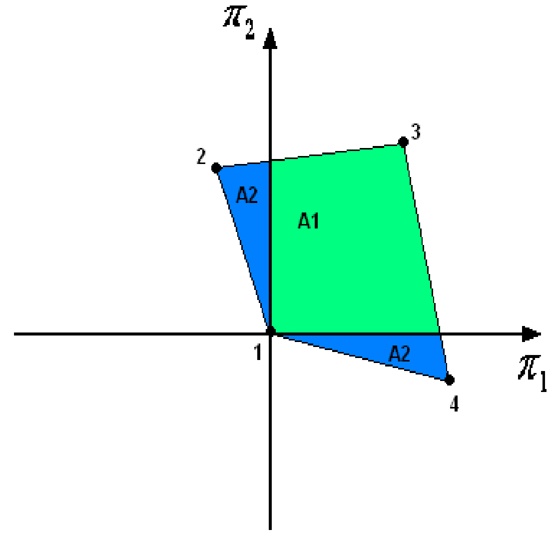


Figure 6. Payoff profile representation for player actions.

The feasible region of the framework is inside a convex hull of, {1.(0,0), 2. $contribution, -(NA * 0.083)$), 3. $(contribution - cost, [(\pi_1(a_1, a_2) + P_{(C)} + P_{(A)})/I] - (NA * 0.083))$, 4. (-$cost, ((\pi_1(a_1, a_2) + P_{(C)} + P_{(A)})/I)$}. Due to the nature of this game, it can exist in an infinite number of equilibriums. Figure 6 represent these equilibriums by taking into account the players strategic actions. Region A1+A2 shows the feasible region of this game. Area A1 shows the enforceable region of the game where the payoffs are positive for any action made by the players. Figure 6 shows that this game can exist with an infinite number of equilibriums. In terms of accuracy, not all of them yield good annotation performance. The payoff profiles inside the area of A2 shows the less-rational game players who are more likely to be cheaters. The payoff profiles located in the region A1 are always positive, and this region contains the players who contribute to the game by providing good annotations. Therefore, when the players are in region A1, the framework temporally accepted their outcomes as good annotations.

3.5 Score Computation

The score computation is another important section of the framework, which has been used for two purposes; firstly to reward the player for his contribution (encouraging them to be more in gaming); secondly to measure player 1's action "a_1", (framework used this information to calculate the payoff functions).

This game used two types of players rewarding schemes. One scheme is used to reward a player for annotating a fully annotated image and the other scheme rewards a player for annotating a non-annotated image. The score computation rewards a player with 100 points for a good annotation. Furthermore, it is designed to decrease the score by 120 points for a bad annotation. A player will be rewarded for annotating a non-annotated image is calculated by taking into account of the player 2's payoff. This is used to represent the confidence level of the framework based on the player 1's contribution and the other related factors. Furthermore, a player will be rewarded by the framework as follows.

$$Player\ 1's\ score = player\ 2's\ payoff * 100 \quad (13)$$

We used player 1's score as a criterion to measure player 2's action"a_2". According to eq.13, player 1's score was calculated based on the confidence level of player 2. However, player 2's action was calculated using a threshold score, which was calculated based on the player 1's contributions to the game. Although player 1 increases his score, the framework keeps a difference of 300 points between the player and threshold score. However, when player cheats, his score will be reduced while keeping the threshold score unchanged. Therefore, for some cheaters, the player 1's score will become less than the threshold score. This strategy reveals the dishonest players in this game, who are less accurate in image annotation.

3.6 Optimizing strategies

We used two optimizing strategies to increase the efficiency of this game. These strategies are used to optimize the payoff profile thus it improve the efficiency of the annotation process.

Enforcing the framework to be in region A1 is one of the ways to optimize the framework. In practice, it is a difficult task to achieve. However, optimization can be achieved to some extent by calculating the player 1's payoff ($\pi_1(a_1, a_2)$) before loading a non-annotated image (this information is always available to the framework before loading images). If the player payoff is less than 0, the framework will not load any non-annotated image for the player. This is because the Nash equilibrium analysis of the framework is designed to eliminate the outcome (payoff) below zero.

Before loading any non-annotated images, the proposed framework ensures that the player score is always greater than the threshold score. This strategy increases the performance in image annotation by not loading any non-annotated image to the player, where the Nash equilibrium analysis will discard the annotation because of $a_1 = 0$.

These two optimizing strategies are used in the framework to improve its efficiency and performance in image annotation.

4. PERFORMANCE MEASURE

The performance of the framework is analyzed as follows: first, the performance is analyzed in the image classification and then the outcomes of the framework were analyzed using a group of 10 game players. These players were used to measure the performance of the combined framework (MM combined with the two-player game model), and the performances of the Bayes approach combined with the two-player game model.

4.1 Performance measure in image classification (SVM)

Due to the use of image classification techniques, the performance of the SVM classifier was measured for a number of concepts. It was trained for 100 images (100 positives and 100 negatives) for each concept butterfly, cougar, tree, building, cloud and tiger.

Table 1. Performance of the SVM classifier

Precision	Butterfly	Cougar	Tree	Building	Cloud	Tiger
CLD	45%	12%	65%	45%	62%	50%
DCD	30%	5%	40%	20%	54%	45%
EHD	45%	12%	40%	65%	73%	53%
Merged descriptor	53%	16%	75%	75%	76%	58%

Table 1 demonstrates the performances of the classifier for a number of concepts. It shows the merge descriptor performs well in image classification than using a single descriptor. A liner fusion strategy was used here to combine all 3 low-level visual descriptors CLD, DCD and EHD to gain the performance in image classification.

Due to computational considerations, the image features have been extracted off-line. This process reduces the computing requirements for the game, thus it makes the game capable of working in any type of computer.

4.2 Player outcome prediction - MM with two-player game model

The performance of the player prediction model is studied for 16 fully annotated and 8 non-annotated images. Figure 7 to 9 show the outcome of the framework for three types of game players such as, a classical cheater (a player who does good annotations in the beginning of the game and then cheats), a random cheater and a true game player. The following figures show a square sign for correct annotations detected by the framework, a bad annotation done by the framework is shown by triangles, and a bad annotation detected by the framework is shown by circles along the player confidence line $P(U)$. The terms RR, RW, WW and WR are the transition probabilities of the MM.

4.2.1 Performance measure of a classical cheater

The performance of a classical cheater is shown by Figure 7 (who used to do good annotations in the beginning and then start cheating). It shows that the proposed framework is able to annotate 2 out of 3 images correctly (before player start cheating). The triangle indicates the point where the framework was unable to predict the miss-behavior of the player (the lack of information in the MM misleads in prediction). Furthermore, the given images from the system have not been annotated by a significant number of players, which is the main reason for failed in detecting bad annotations. At point 13, the framework found a wrong annotation. Therefore, it calculates $P(1-U)P(W|W)$ and $P(1-U)P(R|W)$ probabilities to find the highest probability, which is most likely to be the state of the observation of the next outcome. In this scenario $P(1-U)P(W|W)$ is the highest probability that eventually indicates the framework that the player's next move could be a wrong annotation. Therefore, the framework loads a fully annotated image to the player in the next game round.

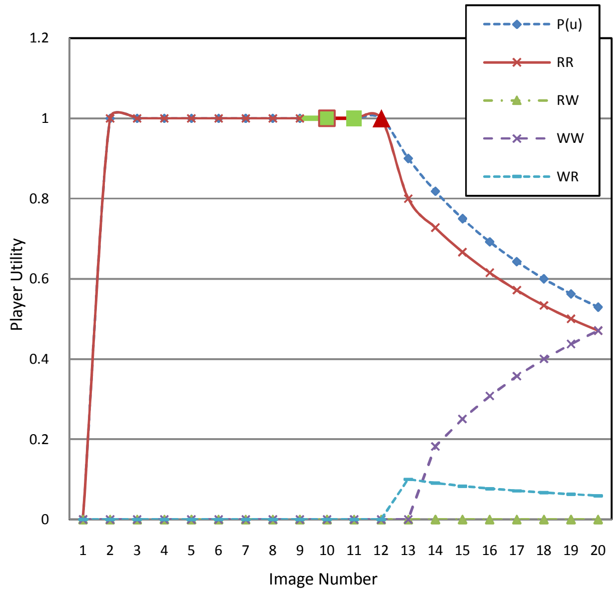


Figure 7. Performance measure of a classical cheater.

4.2.2 Performance measure of a random cheater

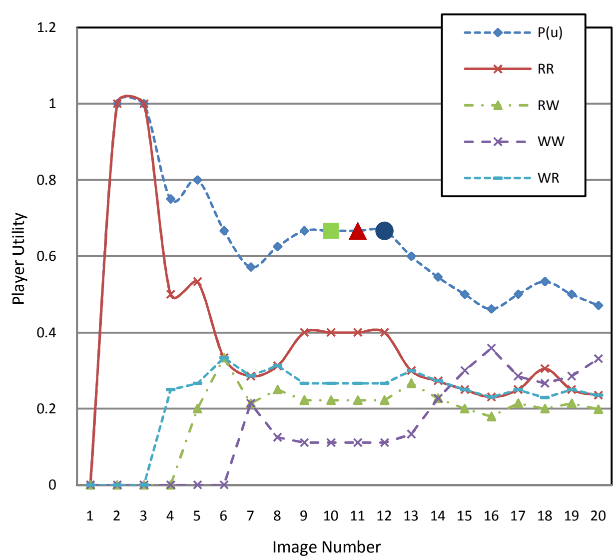


Figure 8. Performance measure of a random cheater.

Random cheaters are the most difficult factor to detect in practice. Therefore, we analyze the performance of the framework for a random cheater, which is shown by Figure 8. In the beginning the player annotated 5 out of 8 images correctly. At point 10, the framework calculates the probability $P(U)P(R|R)$ and $P(U)P(W|R)$ to predict the players next outcome. In this scenario, $P(U)P(R|R)$ is the highest probability that indicates the player's next move could be a good annotation. Hence, the framework loads the player a non-annotated image. However, at point 13, the framework founds that the player is misbehaving. Therefore, it calculates $P(1-U)P(W|W)$ and $P(1-U)P(R|W)$ probabilities to predict the player's next move, where $P(1-U)P(R|W)$ is the highest probability that indicates the player next move could be a good annotation. However, system loads a fully annotated image to the player at point 14, which was generated by the random image generator.

4.2.3 Performance measure of a true player

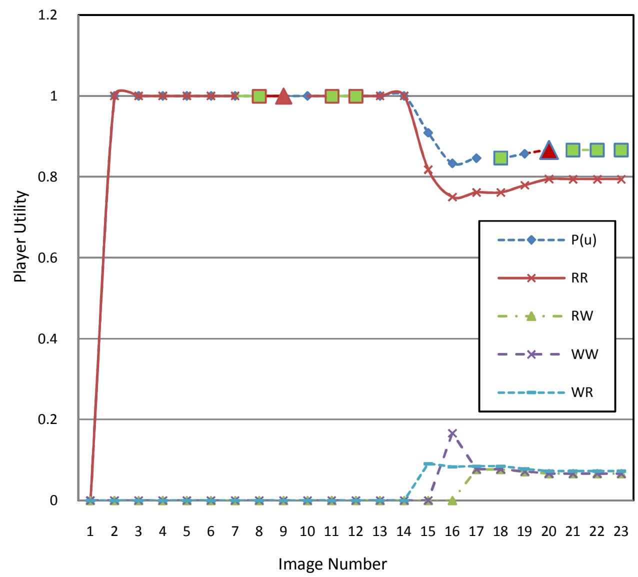


Figure 9. Performance measure of a true game player.

Figure 9 shows the behavior of a true game player. It shows that the proposed framework performed well in this scenario by obtaining good annotations. Furthermore, the player performed well by giving good annotations to the system (the player has been encouraged by the game or interested in collecting game points where the most of the game players are keen on doing [7]). Figure 9 shows the player annotated 7 out of 9 images correctly. However, spelling mistakes and misclassification led the framework to have 2 bad annotations in the game.

4.3 Player outcome prediction by Bayes Theorem

The performance of the Bayes framework has been studied for 8 non annotated and 16 fully annotated images. The Bayes theorem calculates the contribution level of a player by analyzing their annotations made for a series of fully annotated images. This framework predicts a player outcome similar to the MM approach. As an example, let's assume that the player did a good annotation, then the framework calculates the probabilities P(R|R) and P(W|R) to find the state of the highest probability, which is more likely to be the players next outcome. If the probability P(R|R) > P(W|R), the framework assumes the players next move will be a good annotation. Here, we used the same input (annotations) as we used in section 4.2 for testing.

4.3.1 Performance measure of a classical player

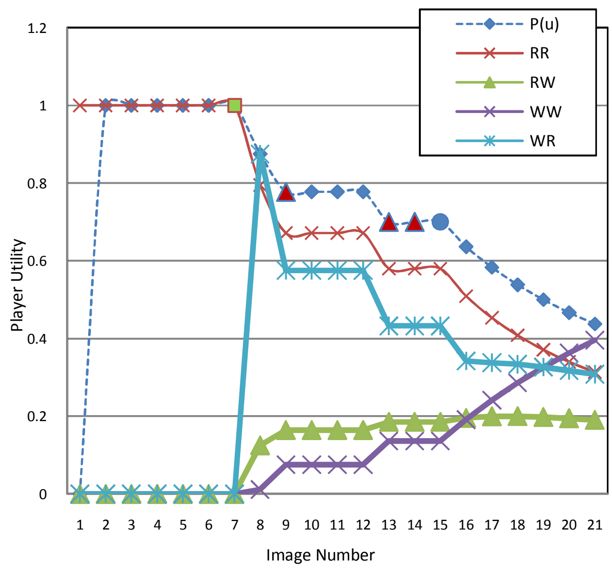


Figure 10. Performance measure of a classical cheater.

Figure 10 shows the performance of the framework for a classical cheater. This player cheats at point 9, where the framework unable to detect. This causes the system to load more non-annotated images. However, at point 15 the two-player game model detects the misbehavior of the player. This causes the framework to load fully annotated images to the player.

4.3.2 Performance measure of a random cheater

It is very difficult to measure the performance of a random cheater. However, Figure 11 shows that the Bayes theorem performed well in this game by detecting random cheaters. It shows, after point 9 the framework did not load any non-annotated images to the player.

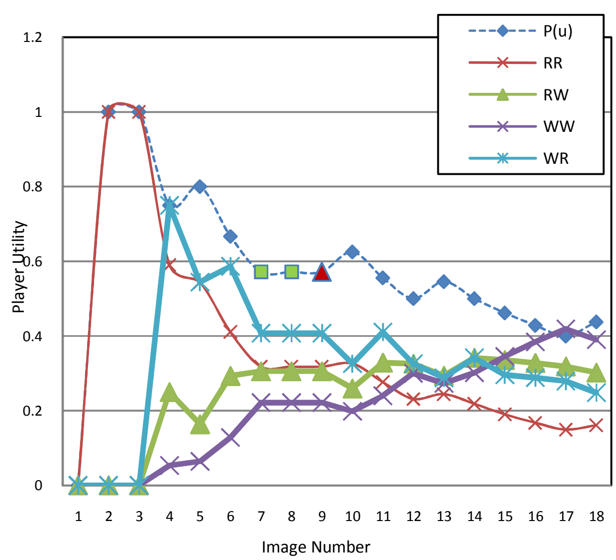


Figure 11. Performance measure of a random cheater.

4.3.3 Performance measure of a true player

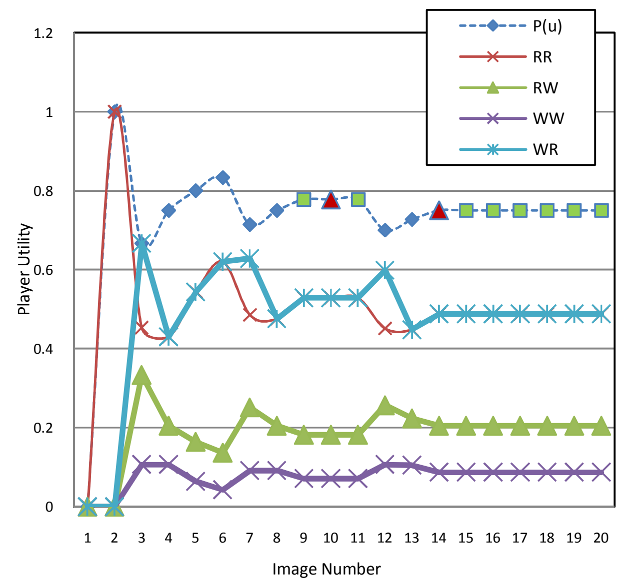


Figure 12. Performance measure of a true game player.

The Bayes confidence level for a true game player is shown by Figure 12. It shows that the majority of images have been well annotated by the framework. Furthermore, the player performed well in image annotation as well. However, spelling mistakes and misclassification led the framework to have some bad annotations in this game session.

4.4 Performance measure, MM vs. Bayes approach along the two-player game model

We studied the performance of the framework for MM and Bayes approach. These approaches were compared with each other to evaluate their performances.

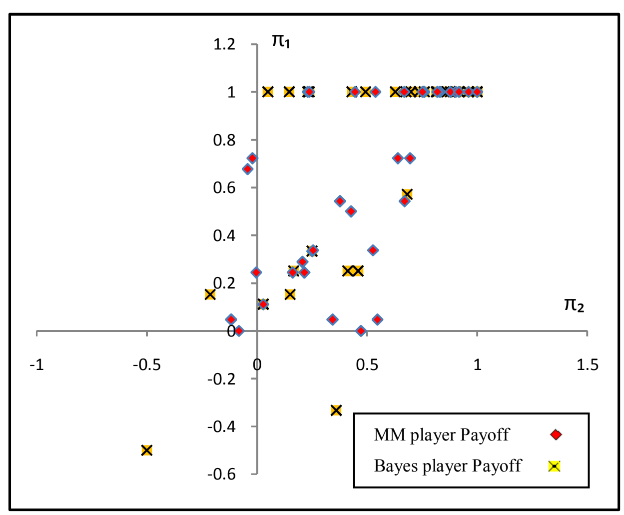


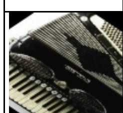
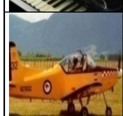
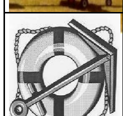
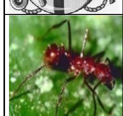
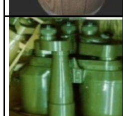
Figure 13. Payoff profile representation (MM vs. Bayes approach).

These experiments show that the MM is the most suitable approach of detecting classical cheaters or true game players and is better than the Bayes model. The Bayes model shows that it is the most capable approach of detecting random cheaters and is better than MM.

Figure 13 shows that the MM detects less-rational game players better than the Bayes model. However, the performance of the Bayes approach is higher in player prediction than the MM approach (Bayes obtains an accuracy of 72.4%, where the MM obtains 68.7%). The main reason for Bayes' success is the fact that most of the players who played this game are random cheaters. They challenged the framework rather than raising their game points. However, in many cases the proposed framework shows that it is capable of detecting less-rational game players by preventing them from generating no useful information.

Table 2 shows the outcome of the framework for 8 non-annotated images (annotated by a group of 10 game players). It shows that the performance of the framework can be improved by adding a spell checker and dictionary mechanisms. Furthermore, this framework shows that it is capable of obtaining a good annotation by using a set of few game players (the annotations eliminated from the framework are not shown in Table 2).

Table 2. Performances of the complete framework for 10 game players

Image	MM				Bayes			
	Annotation	Votes	Av. Payoff (player 1)	Av. Payoff (player 2)	Annotation	Votes	Av. Payoff (player 1)	Av. Payoff (player 2)
	Accordion Dog	6 1	0.5486 0.5428	0.539 0.3768	Accordion F Car	3 1 1	1 1 0.333	0.704 0.917 0.2503
	Plane Computer Cv	4 1 1	0.6476 0.244 1	0.7637 0.1614 0.751	Plane Tub	3 1	1 1	0.9007 1
	Anchor Cat Ac	2 1 1	1 0.5428 1	0.834 0.3768 0.917	Anchor	4	0.8928	0.8622
	Ant D	3 1	0.8333 0.1111	0.5746 0.0281	Ant	4	0.8125	0.6192
	Barrel Barre	1 1	1 0.333	0.2302 0.2503	Barrel S	3 1	3 0.25	0.1423 0.167
	Binocular Xt	2 1	0.8611 0.2888	0.8475 0.2058	Binocular Sun	3 1	0.75 0.1111	0.7192 0.02295
	Fish Cat	2 1	1 0.3375	0.2369 0.2545	Fish D	1 1	1 0.5555	0.2365 0.0281
	Bonsai Bon	2 1	0.6687 0.7222	0.6784 0.6392	Fish D	1 1	1 0.5757	1 0.3220

5. CONCLUSIONS

This paper introduced a novel technique for image annotation: a game used to harness human brain power towards image annotation. The proposed approach is a standalone game that is capable of entertaining and motivating and is used to provide valuable information on image contents. This game combines a number of different key paradigms such as image processing, machine learning and game theoretic approaches to obtain an accurate annotation. To increase the accuracy in annotation, this framework predicts a player outcome. Thereby, it selects the most suitable image for the player (a fully annotated or a non-annotated image). We have proposed two different algorithms for player prediction. The first approach used a MM and second approach used the Bayes theorem. To increase the accuracy in image annotation, this framework is designed as a two-player (virtual) game model. Using this model, the framework aggregates information such as image classification and previous outcomes (annotations made by other players) to filter out the user input (annotations). Using this strategy, the framework eliminates annotations made by less-rational game players. Hence, it improves the accuracy in image annotation.

The results obtained by the proposed framework are promising. They show the MM approach obtains an accuracy of 68.7% in image annotation, where the Bayes model shows an accuracy of 72.4%. However, these frameworks have their own advantages. For example, the MM approach is capable of differentiating classical cheaters and true game players better than the Bayes model. On the other hand, the Bayes model performed well in detecting random cheaters.

Our future work will be mainly concerned with designing a hybrid system to improve the performance in image annotation (combining the MM and Bayes approaches). Furthermore, to improve the accuracy and efficiency of the framework, a dictionary and a spell checker will be introduced in the future.

6. ACKNOWLEDGEMENTS

The research leading to this paper was partially supported by the European Commission under contract FP7-216444, PetaMedia.

7. REFERENCES

[1] Keysers, D., Breuel, T., Renn, M. and Beusekom, J. 2008. Automatic Image Tagging using Community-Driven Online Image Databases, 6th International Workshop on Adaptive Multimedia Retrieval.

[2] Tuulos, V., Scheible, J. and Nyholm, H. 2007. Combining Web, Mobile Phones and Public Displays in Large-Scale: Manhattan Story Mashup. LNCS, Volume 4480, pp 37 – 54.

[3] Foto tagger, A collaborative framework for image tagging, Cogitum co. http://www.fototagger.com/biz_ftonline

[4] [Online]. Available : http://www.xmlweb.info:8080/Collimator3_1/

[5] Von Ahn, L. and Dabbish., L. 2004. Labeling images with a computer game. ACM Conference on Human Factors in Computing Systems (CHI).

[6] Jesus, R., Goncalves, D., Abrantes, A., Correia, N. 2008. Playing Games as a Way to Improve Automatic Image Annotation, Computer Vision and Pattern Recognition Workshops (CVPRW).

[7] [Online]. Available : http://www.youtube.com/watch?v=l0wLQJsgTgk&feature=related.

[8] Puterman, M.L. 1994. Markov Decision Processes: Discrete Stochastic Dynamic Programming. John Wiley & Sons, Inc.

[9] Nasr, A., Bechet, F., Volanschi, A. 2004. Tagging with hidden Markov models using ambiguous tags. Proceedings of the 20th international conference on Computational Linguistics.

[10] Weber, I., Robertson, S. and Vojnovic, M. 2008. Rethinking the ESP Game. Technical Report, MSR-TR-2008-132, Microsoft Research.

[11] Von Ahn, L., Liu, R. and Blum, M. 2006. A Game for Locating Objects in Images. ACM Conference on Human Factors in Computing Systems (CHI).

[12] Ho, C., Chang, T., Lee, J., Hsu, J.Y. And Chen, K. 2009. KissKissBan: A Competitive Human Computation Game for Image Annotation. KDD-HCOMP'09.

[13] Von Ahn, L., Kedia, M. and Blum, M. 2006. Verbosity: A Game for Collecting Common-Sense Facts. ACM Conference on Human Factors in Computing Systems (CHI).

[14] Von Ahn, L.,Ginosar, S., Kedia, M. and Blum, M. 2007. Improving Image Search with PHETCH. Acoustics, Speech and Signal Processing, ICASSP 2007. IEEE International Conference on, Volume 4, Issue , pp15-20.

[15] [Online]. Available : http://www.hotornot.com/.

[16] [Online]. Available : http://images.google.com/imagelabeler/.

[17] Lin, W.S., Zhao, H. V. and Liu, K.J.R. 2009. Incentive Cooperation Strategies for Peer-to-Peer Live Multimedia Streaming Social Networks. IEEE Transactions on Multimedia, Volume.11,No.3, pp396-412.

[18] Manjunath, B.S., Ohm, J.R., Vasudevan, V.V. and Yamada, A. 2001. Colour and Texture Descriptors. Circuits and Systems for Video Technology. IEEE Transactions on Volume 11, Issue 6 (Jun 2001), pp703 - 715

[19] Won, C. S., Park, D.K. and Park, S. "Efficient Use of MPG-7 Edge Histogram Descriptor", ETRI Journal, Volume 24, Number 1, pp23-30, Feb 2002.

[20] Evaggelos, S., Borgne, H.L., Mailis, T., Cooke, E., Avrithis, Y. and Connor, N. 2005. Fusing MPEG-7 Visual Descriptors for Image Classification, Artificial Neural Networks: Formal Models and Their Applications (ICANN).

[21] Gunn. S.R. 1998. Support Vector Machines for Classification and Regression. Technical report, University of Southampton.

[22] Mendelson, E. 2004. Introducing Game Theory and Its Applications, Taylor & Francis Ltd.