

# Final Team Report: A Novel GWAP Disaster Monitoring System

---

Team Hotpot

Changkun Ou<sup>\*</sup>, Yifei Zhan<sup>†</sup> and Zhe Li<sup>‡</sup>

Lecture Human Computation

*Institute for Informatics, Ludwig-Maximilian University of Munich, Germany*

*{changkun.ou<sup>\*</sup>, yifei.zhan<sup>†</sup>, li.zhe<sup>‡</sup>}@campus.lmu.de*

August 1, 2017

## CONTENTS

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Related Works	3
1.1.1	UNICEF	3
1.1.2	Human Computation System and Game With A Purpose	3
1.2	Purpose of the System	4
1.3	Human Contribution to the System	4
<b>2</b>	<b>Functionalities</b>	<b>4</b>
2.1	Functionalities as Seen by a Player	5
2.1.1	Disaster Monitoring Game Introduction	5
2.1.2	Examples	5
2.2	Functionalities as Seen by a Stakeholder	7
2.3	Possible Technology Stack for Implementation	8

<b>3</b>	<b>Design</b>	<b>8</b>
3.1	System Architectures . . . . .	9
3.2	System Components . . . . .	10
3.2.1	Database Fields . . . . .	10
3.2.2	Player Task Generator . . . . .	12
3.2.3	Player Rating Model . . . . .	12
3.2.4	Disaster Evaluation Model . . . . .	17
3.3	Model Initialization . . . . .	18
3.4	Discussion on Image Independent Model . . . . .	19
<b>4</b>	<b>Evaluation</b>	<b>19</b>
4.1	Success Criterias . . . . .	19
4.1.1	Model Evaluation by Simulation . . . . .	19
4.1.2	Issues on Social and Ethical Aspects . . . . .	20
4.2	Limitations of the System . . . . .	21
4.2.1	Evaluation Outdated . . . . .	21
4.2.2	Gameplay and Playability . . . . .	21
<b>5</b>	<b>Conclusions and Future Works</b>	<b>21</b>
5.1	Conclusions . . . . .	21
5.2	Future Works . . . . .	22
	<b>Acknowledgements</b>	<b>23</b>
	<b>References</b>	<b>23</b>

**ABSTRACT** Disaster Monitoring is a challenging problem due to the lack of infrastructures in the disaster areas. This report contributes to Disaster Monitoring with the description of a Game With A Purpose (GWAP), which analyzes tagging results from players from satellite pictures and exports aggregated results to Disaster Monitoring stakeholders. We illustrate our system prototype and implementation technology stack, as well as the mathematical model of the scheme. As justification and evaluation, we prove the correctness of the model, discuss issues caused by this system and possible solutions as well as extensions for future works.

## 1 INTRODUCTION

In this chapter, we will give an introduction about the backgrounds of UNICEF and the related technics used in the project. The simple definition of HC system and GWAPs will be given. The purpose of our HC system and the contribution of human beings will also be talked.

TODO: Fix all section title issue (start with capital letter, all lower case)

### 1.1 RELATED WORKS

#### 1.1.1 UNICEF

The United Nations Children's Fund [Uni94] is a United Nations programme headquartered in New York City that provides humanitarian and developmental assistance to children and mothers in developing countries [Wik17c]. It works in 190 countries and territories to protect the rights of every child. UNICEF has spent 70 years working to improve the lives of children and their families. Defending children's rights throughout their lives requires a global presence, aiming to produce results and understand their effects. In Syrian, the UNICEF works on providing and transporting critical medicine, aid and supplies to the refugees living in the war areas. The challenges UNICEF meet is that there are many hard-to-reach (HTR) and besieged (BSG) areas and the supplies are very hard to be delivered to these zones if the UNICEF have no idea about the real time war situation and the disaster level. It will cost too much for the UNICEF which is just a Non-profit organization if they entirely hire employees to collect the data of war situation. Our work is to design and develop a Human Computation system by GWAP [LJB15].

#### 1.1.2 HUMAN COMPUTATION SYSTEM AND GAME WITH A PURPOSE

TODO: fix here by reintroducing human computation HC and game with a purpose GWAP abbreviation.

**Human Computation (HC) system** is a paradigm for utilizing the human processing power to solve problems that computers cannot yet solve [QB11]. It is the system of computers and

large numbers of humans that work together in order to solve problems that could not be solved by either computers or humans alone [QB09]. Our HC system is a kind of GWAPs, which uses enjoyment as the primary means of motivating participants. One of the challenges in any human computation system is finding a way to motivate people to participate [QB11]. Besides the enjoyment, we will design some interactions between users and our system to make the volunteer users feel honored for their contribution.

## 1.2 PURPOSE OF THE SYSTEM

The users are required to select a **Region Of Interests(ROI)** upon the presented satellite images and tag the ROI from a provided tag list or input their own tag. Anyone can directly participant without registration, but the system will record an ID of each user. Computer Graphics can also be a way to detect and recognize the map images, but it will cost too much time and money in developing recognition algorithms and the best computer graphics algorithm currently can not beat the image recognition ability of human beings. That's the reason we design the HC system to solve the problem.

## 1.3 HUMAN CONTRIBUTION TO THE SYSTEM

The Computer Graphic techniques and Artificial Intelligence grow very fast in recent years, however, it is still a great problem for computers to detect and recognize images accurately and fast. Nevertheless, it is a simple thing for human beings to do it. The HC system for disaster monitoring encourages more Internet users to contribute information to solve the image tagging problem by GWAPs. We developed the Player Rating Model to guarantee the quality of collected information and some interesting feedback and interaction are designed to maintain the enjoyment of players in the game. Users do some image tagging tasks in the game by their computing power and intelligent which are contributed to collect data in the map images.

# 2 FUNCTIONALITIES

In this chapter, we introduce two interactive mockups of our disaster monitoring model. The first mockup is built for game players, which is used for collecting human inputs. The second mockup is built for some collaborative organisations, such as Unicef and some NGOs, who wants to dispatch their rescue teams in disaster area more properly.

## 2.1 FUNCTIONALITIES AS SEEN BY A PLAYER

### 2.1.1 DISASTER MONITORING GAME INTRODUCTION

The idea behind our human computation system is GWAP, that is Game With a Purpose. We have sketched a mockup of image tagging game, which is similar to the ESP game on Artigo.com [WBBL13]. In our game, A player can finish infinity round tasks, a Round task contains  $N$  tagging tasks and a tagging task is to: interpret one picture. Within one round task, the player will see  $N$  pictures. Each time he/she will be asked to tag one of these pictures. At first, the player needs to draw a rectangle area which indicates that he/she has seen some objects in this area. Thoses objects are often considered as a sign of danger or damage. They are mostly like:

- “Rocket Launcher”
- “Armoured vehicles”
- “Tanks”
- “Burning building”
- “Heavy vehicle tracks”
- “sign of explosion”

Each of them has a value of disaster level, which will be used later in our (see chapter 3.2.4) for analysing the disaster level of this region. A submenu list (“Pre-provided item list”) which contains all of these items will pop up after the area has been selected. In this step, the player can decide which one of these items matches and then choose it. The player can also obtain helps from the reference panel which provides the player with examples. Besides, if the player does not find the expected item in the “Pre-Provided item list”, he/she can also create a new tag.

In a tagging task, the player also has a choice to add multiple tags to one picture, or he/she doesn't need to add any tags when he/she thinks that this region is safe.

After finishing one tagging task, the player will be directed to the next task till the end of the game.

### 2.1.2 EXAMPLES

In this section, An example of a user journey is provided in order to illustrate the game process.

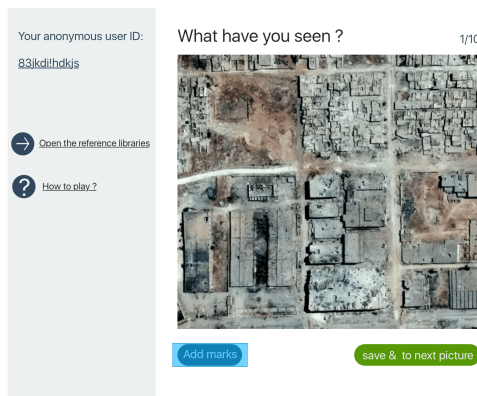


Figure 2.1: Game panel [GH16]

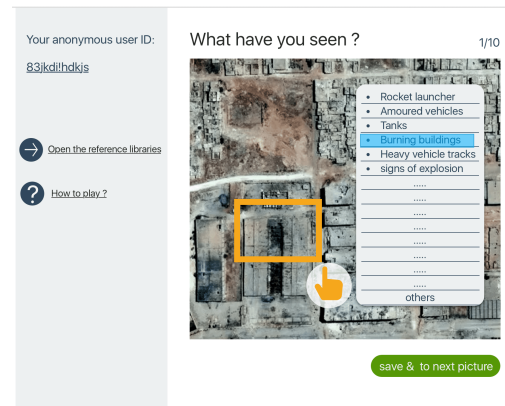


Figure 2.2: Game panel: Tag list [GH16]

Imaging that user A is now at the very first beginning of the game. What he will see, are a side bar on the left side, which contains some information like: user ID, game guide and a reference library, and a main panel on the right side (Figure 2.1). If user A finds a sign of damage or danger in the picture on the right side. He can click the button “add marks” and then he is able to draw a rectangle area which indicates the location of the sign. A selection box will then pop out automatically and user A can select the suited item or add a new tag (Figure 2.2).

As already mentioned above, user A can also add multiple tags to one picture (Figure 2.3).

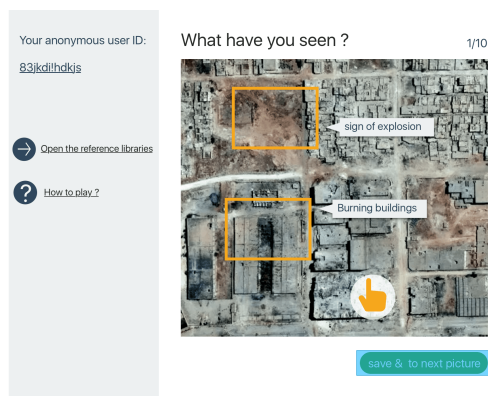


Figure 2.3: Game panel: Multiple tags [GH16]

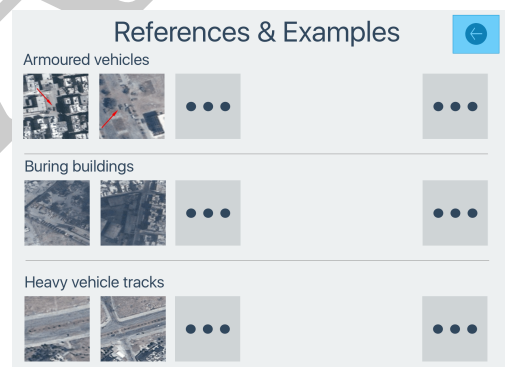


Figure 2.4: Game panel: Reference library.  
Image ©DigitalGlobe, Inc

After user A finishes this tagging task, i.e. tagging the first picture, he can click the button "save and to the next picture" so that he can save the tags and go to the next task. Figure 2.4 illustrates how a reference library can be, whose aim is to help user A to identify different signs of danger or damage.

## 2.2 FUNCTIONALITIES AS SEEN BY A STAKEHOLDER

In our disaster monitoring system, we take image taggings from players as our input. In the next step, we will filter and analyse the data. The final output of our work is a disaster level report in some certain region, which can be used by some organisations, like: Unicef, some NGOs and even governments.

Under this consideration, we also sketched a mockup for this user group, which gives them an overview of the disaster level and to download the report at the same time (see figure 2.5).

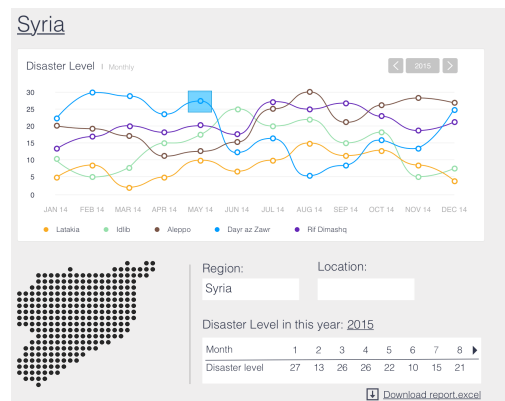


Figure 2.5: disaster level report platform

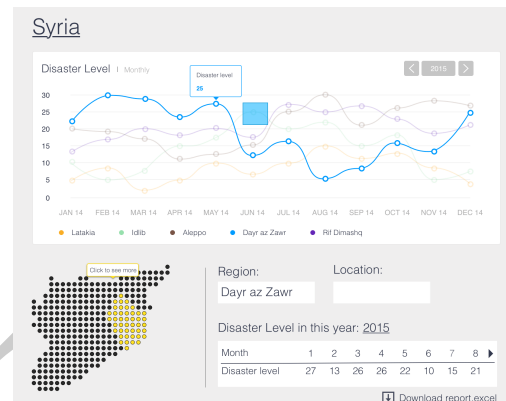


Figure 2.6: disaster level report platform

The platform is composed of a curve chart, a map and some statistics. In the curve chart, each curve shows the yearly disaster level of a smaller region. Like here in figure 2.5, the whole region is Syria and each curve stands for a province in Syria. The user can click the "dot" on curves, which stands for the disaster level of that month. Meanwhile, the corresponding area will be highlighted on the map (figure 2.6).

In the statistic part, the user gets an overview of the yearly disaster level of a region. He/she also have the opportunity to download it. If the user wants to dig deeper, he/she can click on the highlighted area on this map, which will direct him/her to that region (figure 2.7).

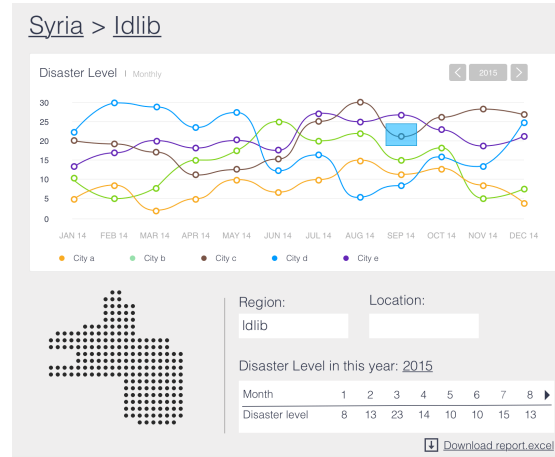


Figure 2.7: Disaster Level Report Platform

In figure 2.7, the user is directed to the province Idlib and now each curve stands for a smaller region in this province, in other words, different cities in Idlib.

### 2.3 POSSIBLE TECHNOLOGY STACK FOR IMPLEMENTATION

In this section, we will discuss the of the future implementation of our system.

We suggest to build our system sketched above as a web-based application so that we do not need to care about the platform. Our system should be capable of easily accessing by both android users and IOS users. We suggest to choose **Polymer**, a Google front-end framework, as our front-end tool and as back-end tool, we suggest to use **Nodejs** and **Python**. For our database, we suggest then to choose **MongoDB**, since it is more suitable for a web-application.

## 3 DESIGN

In this chapter, we describe the overall design of our disaster monitoring backend system in details. Firstly, we propose our system architecture of this disaster monitoring; Then we specify and justify our most critical system components such as databases design, **Player Task Generator (PTG)**, **Player Rating Model (PRM)** as well as **Disaster Evaluation Model (DEM)**.

With these components, model and databases, the disaster monitoring system can handle common problems in HC system, such as cold start, malicious player detection, etc. It is also expandable, portable and can be easily applied to any other same image selection and tagging based HC system in different areas.



### 3.1 SYSTEM ARCHITECTURES

Figure 3.1 illustrate the overall disaster system design. The system databases are composed of two different type of databases. The first database **PlayerDB** combines **TrustedDB** and **UntrustedDB** where persistent the player property and raw tagging inputs whether the overall result is reliable or not. The second database is called **ResultDB** where persistent the reliable player inputs.

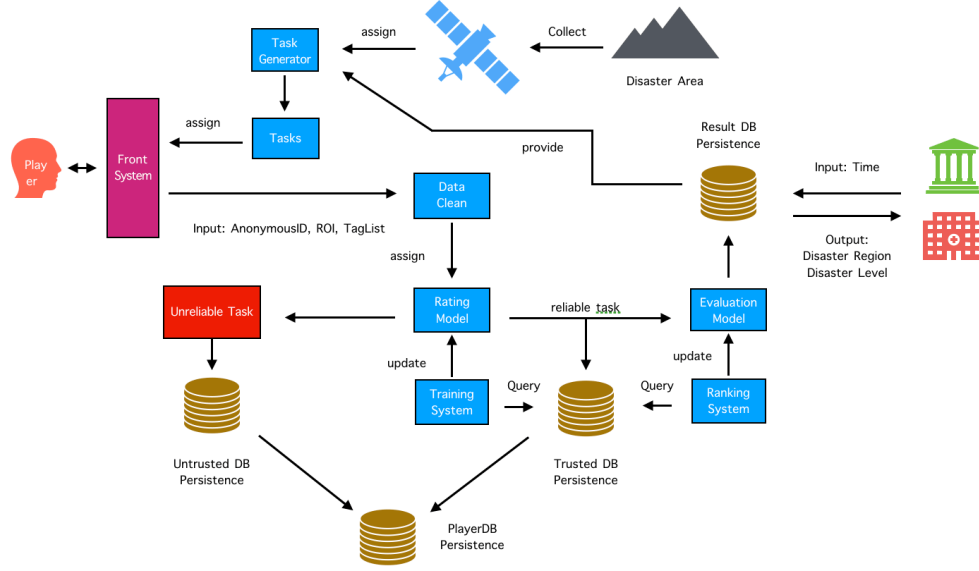


Figure 3.1: System Design Overview

For this architecture design, one can simplify the overall data flow into tree main steps that describes as follows:

- Step 1. Player task generating: We propose the **PTG** that combines trusted results from TrustedDB, and separate new images from satellite, then assign this binding to the future players.
- Step 2. Malicious player detection: A reliable player shall pass the malicious detection algorithm (describe in algorithm 1) inside the **PRM**. Once the player is not a malicious player, then system will mark all the results from this player as a reliable result and then send it into next step.
- Step 3. Evaluating disaster level: the system reuse the reliable player inputs into **DEM** and calculate the disaster level of the monitoring region as well as persistent it in the second database ResultDB.

After these three main steps, stakeholders are able to retrieve monitoring results from the database ResultDB.

## 3.2 SYSTEM COMPONENTS

### 3.2.1 DATABASE FIELDS

We describe the system database PlayerDB fields as well as the fields of database ResultDB first in listing 1 and 2.

In this disaster monitoring system, our participants **do not need to register accounts**, and the system backend shall **generate and assign an *player\_id* to each player** according to the user scenario (such as IP address, network status, system information et cetera). This function significantly accelerate player to participate in this game. Thus, the **PlayerDB** stores the *player\_id* to detect the same players if they participate next time. The player will accomplish different game tasks; each task result shall store in the tasks filed.

```
1 [
2   {
3     "player_id": "E3A6F124-4A6C
4     -4C6E-B7F1-F8BC9A7381CC",
5     "tasks": [
6       {
7         "image_id": "3A21E99E-
8         F074-454B-A590-8
9         D8C5ABD8E77",
10        "image_at": "2017-07-31
11        11:28:40",
12        "reliable": true,
13        "ROIs": [
14          {
15            "x": 103, "y": 121,
16            "height": 56,
17            "width": 78,
18            "tags": ["burning
19            building", "explosion"]
20          }
21        ]
22      }
23    ]
24  }
```

Listing 1: Example of Player Database Data

```
1 [
2   {
3     "region_id": "FBEB6204-0B94
4     -4811-94F0-9DDC5FBBE6D8",
5     "history": [
6       {
7         "image_id": "3A21E99E-
8         F074-454B-A590-8
9         D8C5ABD8E77",
10        "image_at": "2017-07-31
11        11:28:40",
12        "ROIs": [
13          {
14            "x": 103, "y": 121,
15            "height": 56,
16            "width": 78,
17            "tags": ["burning
18            building", "explosion"]
19          }
20        ]
21      }
22    ]
23  }
```

Listing 2: Example of Results Database Data

In the **ResultDB**, a *region\_id* is unique and assigned by our system. A region has its monitoring history and composed by its separate images, the only difference between PlayerDB and ResultDB is ResultDB only stores reliable data (no *reliable* field) and images organized by its related region.

To explain other fields and establish our models, we describe few basic definitions for the system models first.

**Definition 3.1.** The **Region of Interests (ROI)** is an indicator that represents the one of the selected two dimensional regions from player. The  $i$ -th ROI from player  $p$  in image  $k$  at image creation time  $t$  is denoted by  $ROI_{p,i,k,t}$ .

Considering image  $k$  **implies** its creation time  $t$  (an image always has its creation time), for convenience, we usually **simplify**  $ROI_{p,i,k,t}$  to  $ROI_{p,i,k}$ . For instance, figure 3.2 shows few examples of ROI on different images.

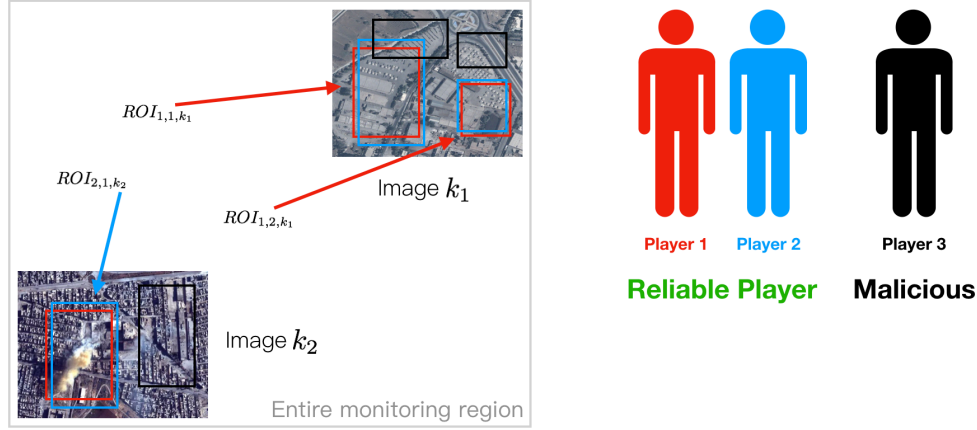


Figure 3.2: Examples of Region of Interests (ROI)

With this definition, our system players are able to select ROIs for each image as well as capable of select tags for each ROI. Thus, the *tasks* field in PlayerDB is an array object, stores each player image result with an assigned *image\_id*. Each object in the *tasks* array has a field *reliable*, which indicates the reliability for this object task; Each object also contains a *ROIs* field, which is an array object that contains the player inputs for this object image; Each ROI object in the *ROIs* field has four properties that describes the ROI geometric location: *x*, *y*, *height*, *width*, and also a *tags* array field that describes the input tags for this image from this player.

For *tags* field, game players select the related tags for each ROI, and stores in this array. **Each tag can only selects once, and players allow to input their own new tags for this ROI.** Then, We define the ROI tag vector for the model design:

**Definition 3.2.** Assuming the database stores  $n$  different tags  $tag_1, tag_2, \dots, tag_n$  for a certain image  $k$ , the **Tag Vector**  $T_{p,i,k}$  of  $ROI_{p,i,k}$  (the  $i$ -th ROI in image  $k$  of player  $p$ ) is a vector that is donated by the following formula:

$$T_{p,i,k} = (|tag_1|, |tag_2|, \dots, |tag_n|) \quad (3.1)$$

where

- $tag_i$  is the  $i$ -th tag;
- $n$  is the number of tags;
- $|tag_i|$  is the count of  $tag_i$  in a player task object.

Note that due to each tag can only select once, the components of tag vector is **either 1 or 0** in reality. For instance, for a certain image  $k$ , there are 5 different tags  $tag_1, tag_2, tag_3, tag_4, tag_5$  were inputted by our game player. Assuming player  $p$  selects the first ROI and inputs tags for  $ROI_{p,1,k}$ :  $\{tag_1, tag_2, tag_3, tag_4\}$ , player  $q$  selects the first ROI and inputs tags for  $ROI_{q,1,k}$ :  $\{tag_1, tag_3, tag_4, tag_5\}$ . Then tag vector  $T_{p,1,k}$  of  $ROI_{p,1,k}$  is  $(1, 1, 1, 1, 0)$  and tag vector  $T_{q,1,k}$  of  $ROI_{q,1,k}$  is  $(1, 0, 1, 1, 1)$ .

### 3.2.2 PLAYER TASK GENERATOR

The **PTG** combines task images from satellite and ResultDB. A player task contains  $2n$  different images in random order, that  $n$  images are the untagged new satellite images and  $n$  images are tagged images of ResultDB, which means **PTG** contains two generating steps:

- Step 1. **PTG** shall split a monitoring region into small pieces of images, and also assign a unique **image\_id** for each piece (The reason is discussed in section 4.1.2 for the leakage of data).
- Step 2. **PTG** shall retrieve tagged images from ResultDB. Then combine all images as a user task assign to a new upcoming player.

### 3.2.3 PLAYER RATING MODEL

This subsection describes the **PRM** as well as its detection algorithm inside our Disaster Monitoring system. The **PRM** is responsible for detect malicious players regarding their game task inputs.

PageRank was first proposed by Lary Page [PBMW99] and applied to social analysis in [BL01]. It is commonly used for expressing the stability of physical systems and the relative importance, so-called **centralities**, of the nodes of a network. We transfer the basic idea of centralities of a network and use eigenvalue as the trust value for each player to distinguish malicious players and reliable players.

We establish the model in image dependent perspective. For a certain image  $k$ , considering a directed **Player Rating Graph (PRG)** between players who tagged the image  $k$ . Each player is a node of PRG, as illustrated in figure 3.3.

**Definition 3.3.** Assuming the database stores  $n$  different tags  $tag_1, tag_2, \dots, tag_n$  in the system, The **System Weight Vector**  $v = (p(tag_1), p(tag_2), \dots, p(tag_n))$  of all tags can be calculated by the

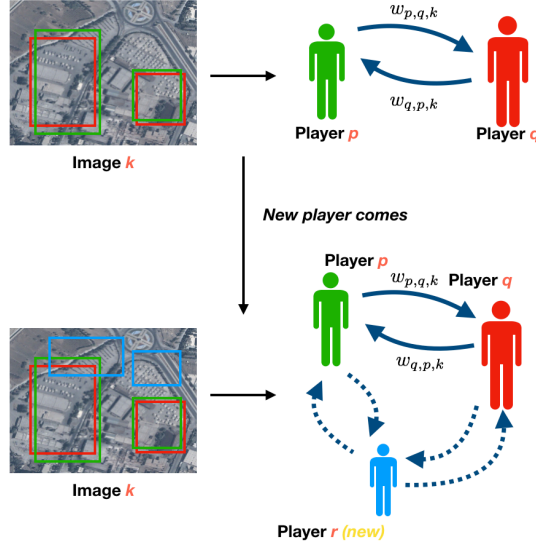


Figure 3.3: Player Rating Model

following equation 3.2:

$$p(\text{tag}_i) = \frac{|\text{tag}_i|}{\sum_{j=1}^n |\text{tag}_j|} \quad (3.2)$$

where

- $n$  is the number of current exist tags;
- $|\text{tag}_i|$  is the occurrence of  $\text{tag}_i$ .

The **Image Weight Vector**  $v_k =$

To go so far as to this system, our player have two different type of inputs: **the ROI, and its Tag Vector**. To define the PRG edge weight, we introduce two input measurements in the subsequent definition 3.4 and 3.5.

**Definition 3.4.** The **Players ROI Matching Ratio (PRMR)** is an importance measurement that measures the proportion of two different ROI intersection surface from player  $p, q$  and the ROI surface from player  $p$  in a certain image  $k$ , which is donated by the following formula:

$$\text{PRMR}(p, q, i, j, k) = \frac{|ROI_{p,i,k} \cap ROI_{q,j,k}|}{|ROI_{p,i,k}|} \quad (3.3)$$

where

- $ROI_{p,i,k}$  is the  $i$ -th selected ROI from player  $p$ ;
- $|ROI_{p,i,k}|$  is the surface of  $ROI_{p,i,k}$ ;

**Theorem 1.**

$$0 \leq \text{PRMR}(p, q, i, j, k) \leq 1. \quad (3.4)$$

*Proof.* According to the definition of ROI,  $|ROI_{p,i,k} \cap ROI_{q,j,k}|$  can archive its maximum value only and only if  $ROI_{p,i,k} = ROI_{q,j,k}$  as well as its minimum value only and only if  $ROI_{p,i,k}$  has no intersection with  $ROI_{q,j,k}$ . Thus:

$$0 = \frac{0}{|ROI_{p,i,k}|} \leq \text{PRMR}(p, q, i, j, k) \leq \frac{|ROI_{p,i,k} \cap ROI_{p,i,k}|}{|ROI_{p,i,k}|} = \frac{|ROI_{p,i,k}|}{|ROI_{p,i,k}|} = 1.$$

□

**Definition 3.5.** The **Players Input Tag Correlation (PITC)** is an importance measurement that measures the proportion of the covariance of two different **Tag Vector**  $T_{p,i,k}, T_{q,j,k}$  from player  $p, q$  and the covariance of  $T_{p,i,k}$  from player  $p$  with itself under the **Weight Vector**  $v_k$ , which is donated by the following formula:

$$\text{PITC}(p, q, i, j, k) = \frac{\text{Cov}(T_{p,i,k}, T_{q,j,k}; v_k)}{\text{Cov}(T_{p,i,k}, T_{p,i,k}; v_k)} \quad (3.5)$$

where  $\text{Cov}(X, Y; w)$  is the weighted covariance between  $X$  and  $Y$ , which donated by:

$$\text{Cov}(X, Y; w) = \frac{\sum_{i=1}^n w_i (x_i - \frac{1}{n} \sum_{i=1}^n w_i x_i) (y_i - \frac{1}{n} \sum_{i=1}^n w_i y_i)}{\sum_{i=1}^n w_i} \quad (3.6)$$

with  $X = (x_1, x_2, \dots, x_n)$ ,  $Y = (y_1, y_2, \dots, y_n)$ ,  $w = (w_1, w_2, \dots, w_n)$ .

Note that

1. The definition of PRMR and PITC share the same intent for measuring asymmetric importance between player  $p$  and player  $q$  (how  $p$  think of  $q$ );
2. The definition of PRMR is inspired by a wide-used computer vision criteria, the so called Intersection over Union (IoU), also as known as Jaccard Index [RV96, Wik17a], which is a statistic used for comparing the similarity and diversity of sample sets. However, in our case, we only divided by to guarantee the asymmetric property for directed graph weight;
3. The definition of PITC is inspired by a weighted Pearson Correlation Coefficient [Wik17b], which is a measure of the linear correlation between two variables. In our case, with the same intent of PRMR, we **drop** the part of covariance of player  $q$  in denominator to guarantee the asymmetric property for directed graph weight;
4. The PRMR and PITC both **are not metrics** due to  $\text{PRMR}(p, q, i, j, k) \neq \text{PRMR}(q, p, i, j, k)$  as well as  $\text{PITC}(p, q, i, j, k) \neq \text{PITC}(q, p, i, j, k)$ .

**Theorem 2.**

$$-1 \leq \text{PITC}(p, q, i, j, k) \leq 1. \quad (3.7)$$

*Proof.* We know that the weighted Pearson Correlation Coefficient [Wik17b] lies on  $[-1, 1]$ , i.e.

$$-1 \leq \frac{\text{Cov}(T_{p,i,k}, T_{q,j,k}; v_k)}{\sqrt{\text{Cov}(T_{p,i,k}, T_{p,i,k}; v_k) \text{Cov}(T_{q,j,k}, T_{q,j,k}; v_k)}} \leq 1$$

To prove equation 3.7, we have to show:

$$\begin{aligned} & \frac{\text{Cov}(T_{p,i,k}, T_{q,j,k}; v_k)}{\text{Cov}(T_{p,i,k}, T_{p,i,k}; v_k)} \\ & \leq |\text{Cov}(T_{q,j,k}, T_{q,j,k}; v_k) \sqrt{\text{Cov}(T_{p,i,k}, T_{p,i,k}; v_k) \text{Cov}(T_{q,j,k}, T_{q,j,k}; v_k)}| \leq 1 \end{aligned} \quad (3.8)$$

and

$$\begin{aligned} & \frac{\text{Cov}(T_{p,i,k}, T_{q,j,k}; v_k)}{\text{Cov}(T_{p,i,k}, T_{p,i,k}; v_k)} \\ & \geq -|\text{Cov}(T_{q,j,k}, T_{q,j,k}; v_k) \sqrt{\text{Cov}(T_{p,i,k}, T_{p,i,k}; v_k) \text{Cov}(T_{q,j,k}, T_{q,j,k}; v_k)}| \geq -1 \end{aligned} \quad (3.9)$$

Then we need to show:

$$0 \leq \text{Cov}(T_{p,i,k}, T_{p,i,k}; v_k) \text{Cov}(T_{q,j,k}, T_{q,j,k}; v_k)^3 \leq 1 \quad (3.10)$$

Considering  $T_{p,i,k}, T_{q,i,k}$  are described in general, with equation 3.6, we only need to show ( $s$  is an vector components index instead of exponential):

$$0 \leq \text{Cov}(T_{p,i,k}, T_{p,i,k}; v_k) = \frac{\sum_{s=1}^n v_k^s \left( T_{p,i,k}^s - \frac{1}{n} \sum_{s=1}^n v_k^s T_{p,i,k}^s \right)^2}{\sum_{s=1}^n v_k^s} \leq 1 \quad (3.11)$$

According to the definition of Tag Vector and Weight Vector, the components of  $T_{p,i,k}$  are either 1 or 0, the components of  $v_k$  lies on  $[0, 1]$ , then we have:

$$0 \leq \left( T_{p,i,k}^s - \frac{1}{n} \sum_{s=1}^n v_k^s T_{p,i,k}^s \right)^2 \leq 1 \quad (3.12)$$

Therefore,

$$0 = \frac{\sum_{s=1}^n v_k^s \cdot 0}{\sum_{s=1}^n v_k^s} \leq \text{Cov}(T_{p,i,k}, T_{p,i,k}; v_k) \leq \frac{\sum_{s=1}^n v_k^s \cdot 1}{\sum_{s=1}^n v_k^s} = 1 \quad (3.13)$$

which proves equation 3.7.  $\square$

**Definition 3.6.** For a certain image  $k$ , the edge weight of the PRG from player  $p$  to player  $q$  is donated by the formula 3.14:

$$w_{p,q,k} = \sum_{j=1}^n \sum_{i=1}^m (\text{PRMR}(p, q, i, j, k) (\text{PITC}(p, q, i, j, k) + 2)) \quad (3.14)$$

To calculate the eigenvalue of the adjacency matrix of PRG, one can use the normalized adjacency matrix through the following formula 3.15:

$$A_k = (a_{p,q,k}) = \left( \frac{w_{p,q,k}}{\sum_q w_{p,q,k}} \right) \quad (3.15)$$

where  $k$  is the image indicator.

**Theorem 3.** *The normalized adjacency matrix  $A_k$  of PRG of a certain image  $k$  is irreducible, real, non-negative, column-stochastic, and diagonal element being positive.*

*Proof. Irreducibility:* For a certain image  $k$ , the PRG is strong connected because the player who selected ROIs in image  $k$  has a direct connection to any other player who also selected ROIs in image  $k$  (the edge weight is well defined according to equation 3.14). Thus, since  $A_k$  is an normalized strong connected PRG adjacency matrix, which proves  $A_k$  is irreducible.

**Real elements:** Each part of the equation 3.14 are real number. Thus, of course, the matrix  $A_k$  elements are calculated by equation 3.15 that are real elements.

**Non-negative elements:** We only need to prove  $TV_i$ ,  $\frac{ROI_i \cap ROI_j}{ROI_i}$  and  $2 - \frac{Cov(T_i, T_j; v)}{Cov(T_i, T_i; v) \times Cov(T_j, T_j; v)}$  are non-negative respectively.  $TV_i$  is the eigenvalues of normalized graph adjacency matrix, thus the codomain of  $TV_i$  lies (0, 1]; For MAR, its range is obviously from 0 to 1, which lies [0, 1]; For  $2 - \frac{Cov(T_i, T_j; v)}{Cov(T_i, T_i; v) \times Cov(T_j, T_j; v)}$ , the Pearson Correlation lies on [-1, 1], then this part lies on [1, 3]. Three parts are non-negative.

**Positive diagonal elements:** The diagonal elements can be formalized by follows:

$$w_{ii} = \sum_{ROI \in ROIs} \left( \frac{ROI_i \cap ROI_i}{ROI_i} \left( \frac{Cov(T_i, T_i; v)}{Cov(T_i, T_i; v) \times Cov(T_i, T_i; v)} + 2 \right) \right) = \sum_{ROI \in ROIs} TV_i > 0$$

**Column stochastic:** according to the definition of matrix  $A$ , the sum of the column elements are:

$$\sum_i \frac{w_{ij}}{\sum_j w_{ij}} = \frac{\sum_i w_{ij}}{\sum_j w_{ij}} = 1$$

□

We have proved the existence and uniqueness of eigenvalues of normalized PRG adjacency matrix; one can use the corresponding eigenvalues to represent the trust value of players. Thus, we have:

**Definition 3.7.** A **Trust Value**  $TV_i$  of player  $i$  represents by the  $i$ -th eigenvalue of normalized PRG adjacency matrix  $A$ .



This definition can represent the rating score from  $i$  to  $j$ . With the trust value of players, we propose our classification algorithm:

---

**Algorithm 1:** Player Classification Algorithm

---

**input** : anonymous IDs, TVs  
**output**: (anonymous\_id, isReliable)  
 Calculate  $TV_{new}$  as the trust value of player *new* ;  
**if**  $TV_{new} \geq \frac{1}{|players|} \sum_{i \in players} TV_i$  **then**  
   | return (anonymous\_id, true)  
**else**  
   | return (anonymous\_id, false)  
**end**

---

In this algorithm, the criterion of classifying new players performs the action that the trust value of a new player should not be less than the mean value of overall trust value of players, which means the tagging performance of new player should not worth than result performance of former players.

Terefore in short, the input and output Data Model of PRM are as follows. For input: (anonymous\_id, area\_id, time, ROIs, tags); For model output: (anonymous\_id, TV).

### 3.2.4 DISASTER EVALUATION MODEL

For an area at time  $t$ , we address the **Disaster Evaluation Model (DEM)** via disaster level definition as follows:

**Definition 3.8.** The **Disaster Level (DL)** of a monitor region is calculated by each area components:

$$DL = \sum_{area \in region} DL_{area}$$

where  $DL_{area}$  is calculated by its corresponding tag vector:

$$DL_{area} = \sum_{i=1}^n v_i \times |tag_i|$$

with  $n$  is the number of current exist tags, and  $|tag_i|$  is the occurrence of  $tag_i$  in the corresponding area.

System like ESP [VAD04], ARTigo [WBBL13] has proved that human inputs are valuable and useful.

Note that sometimes player carries new tags for our system, we also address a solution for this issue via the following steps:

- When a player carries predefined tags: Trivial;

- When a player carries new tags: Directly drop, it is an unreliable result;
- When a player carries predefined tags and also new tags: calculate the trust value without new tags; merge and update all weight vector  $\nu$  via formula 3.14 if the player is reliable, otherwise drop and mark the result is unreliable.

With this definition 3.8, we can calculate the disaster level for a monitoring region. To sum up, the input and the output Data Model of DEM address as follows. For input: (time), (area\_id) or (area\_id, time); For output: (area\_id, time, disaster\_level).

### 3.3 MODEL INITIALIZATION

A cold start of such a system is a common problem in human computation system that is avoided by hiring people to play or learn as long as the number of users or the quantity of data is insufficient. In our system, we have two different cold start problem.

The first cold start problem appears in the PTG. To initialize the whole system, we need to address an initial trusted group for PTG; they shall tagging enough initial trusted result for PTG and then assign to new upcoming players. When a new player is reliable, then the result of this player will become reliable. Meanwhile, the trusted group and available dataset grow larger with this step repeatedly, as shown in figure 3.4.

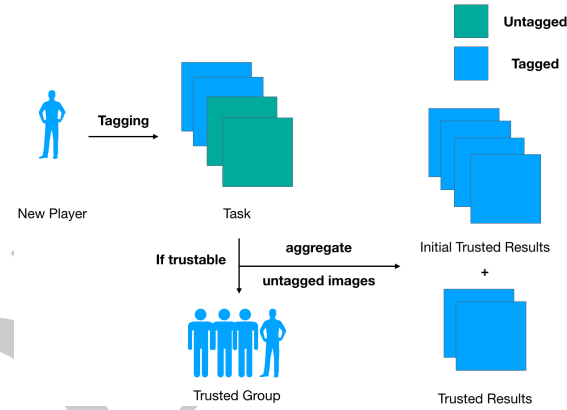


Figure 3.4: Cold Start of PTG

The second cold start problem appears in PRM. According to the definition 3.14 of PRG, the weight of PRG was defined by the trust value of all players. Nevertheless, the initial trusted group has no trust value. Thus we need an initial value for  $TV$ . Note that  $TV_i$  is in between of 0 and 1, thus:

$$TV_i^{\text{init}} = \frac{1}{|\text{players}^{\text{init}}|}$$

with  $|\text{players}^{init}|$  is the number of initial trusted group.

### 3.4 DISCUSSION ON IMAGE INDEPENDENT MODEL

## 4 EVALUATION

This chapter discusse the evaluation criteria as well as the issues and solutions on social and ethical aspects. Surely, our system still has limitations, we then discusse the most important three limitations within our system: outdated evaluation, information loss and playability.

### 4.1 SUCCESS CRITERIAS

#### 4.1.1 MODEL EVALUATION BY SIMULATION

Malicious player detection is a classification problem. One can generate random data and test the performance of PRM through accuracy and recall, even ROC curve [HM82]. TODO: explain here.

The click behavior has been researched for years and address by FFitts Law [BLZ13]. It modeled and proved the distribution of click behavior for a certain click goal point is a normal distribution. Thus, with probablistic view, the top left corner of ROI exists, then the user click selection for this point should follows normal distribution, as shown in figure 4.1. TODO: explain precisely. also justify the start point is not important.

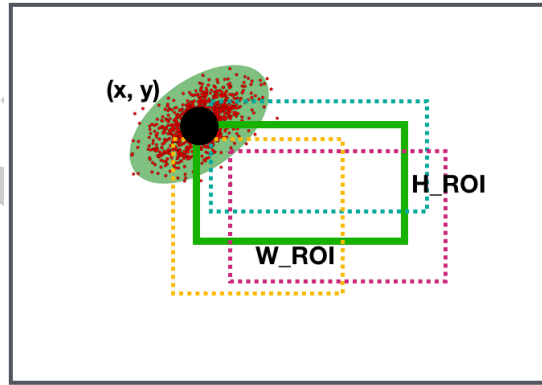


Figure 4.1: Data Simulation

Therefore, to generate ROIs, let  $(x, y)$  is the player ROI start point,  $(H_{ROI}, W_{ROI})$  is the height and width pair of this ROI, then we generate the random dataset for these variables by a given parameter  $\delta$ :  $(x, y) \sim (x + N(0, \delta), y + N(0, \delta))$ ,  $(H_{ROI}, W_{ROI}) \sim (H_{ROI} + N(0, \delta), W_{ROI} + N(0, \delta))$ . To generate tags, we propose randomly pick random number of tags.

Then one can perform this random dataset on our system to evaluate the classification accuracy and recall rate to evaluate the overall performance of this system, which gives the theoretical evaluation results.

#### 4.1.2 ISSUES ON SOCIAL AND ETHICAL ASPECTS

**Intellectual Property Rights:** Our disaster monitoring HC system is an non-commercial project which invites volunteers to contribute to it. It is used to help the UNICEF and other non-profit organizations deliver supplies in disaster areas safer. So the IPR on the annotations should not belong to the volunteer users. In an appropriate manner, user will be informed that his or her contribution to the system is volunteering and the produced data will be used only for monitoring disasters in Syria.

**Leakage of data:** We cut big satellite image into small segmentations in our HC system to prevent data leakage to ordinary users, but the **data security** (TODO: explain here) is still very important and needs to be considered with high priority. For example, the backend database should not be entirely accessible to every employee of UNICEF. Each of them can only visit the part of data that they need at present. This principle is made to guarantee that data will not be wrongly used and will be included in our future work.

**Information Loss** We cut big region images into small fragements areas to prevent leakage of data. But this method will cause some information loss problem if some important ROIs are located at the intersection of two dividing lines. A possible solution for this limitation is to consider new regions that contains the loss informations, as shown in figure 4.2. Note that this solution that only increases the number of region pictures, it does not influence any model and system design.

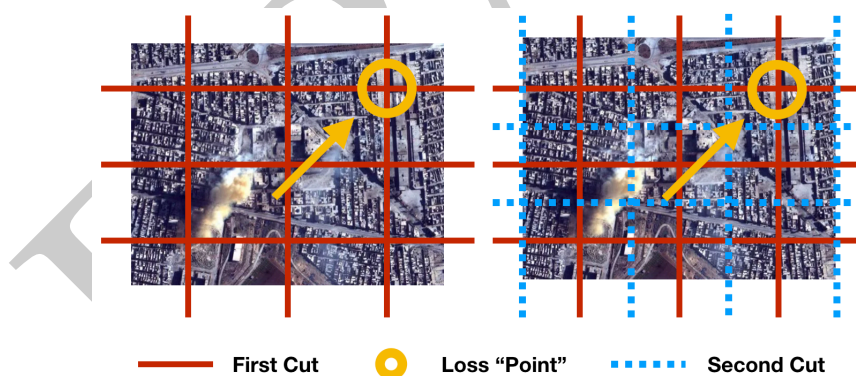


Figure 4.2: **Information Loss:** TODO: fix here. Information loss may occurs on the intersection lines, as shown on the left; a possible solution is to provide additional regions (blue rectangles) for these special area as shown on the right.

## 4.2 LIMITATIONS OF THE SYSTEM

### 4.2.1 EVALUATION OUTDATED

A limitation occurs in our social network based model is each disaster level evaluation get invalid if the region image is outdated. We assume the satellite monitors an area and take pictures between intervals. However, our evaluation the model only calculate the disaster level at a unique moment, which means the disaster level need transvaluation when a new image comes out. If our player is not enough so that the region images always have to wait for new evaluation, then the disaster level will never be calculated.

A possible solution is to consider the region disaster level history as a time series. Then we can apply some prediction method for it. For instance, we have time series:  $(t_1, t_2, t_3, \dots, t_n)$  and its corresponding disaster level:  $(DL_1, DL_2, DL_3, \dots, DL_n)$ . Then we can use these time series to predict the disaster level at time  $t_{n+1}$ .

At the same time, we also have the historical data of trust value of a player. We can also use time series prediction to predict the player's trust value. But in all of these, the time series of disaster level is not stationary but the time series of trust value is stationary.

TODO: Fix here by given few prediction method suggestions.

### 4.2.2 GAMEPLAY AND PLAYABILITY

The GWAP collects satellite photos of disaster areas. But even if in the disaster areas, not every part of the areas has a disaster. Most parts of the earth are lake, forest, desert and so on, which means the users may meet the situation that there is no available ROI in several continuous rounds. Obviously, it will decrease the playability and enjoyment of the game. Our system is just a very simple tagging game at present, users can not get enough enjoyment they want in it. And it is too reliant on the unpaid volunteers to donate their time to contribute information. We should make the system more interesting and appealing in the future work.

## 5 CONCLUSIONS AND FUTURE WORKS

### 5.1 CONCLUSIONS

In this report, we proposed a comprehensive design of a human computation system for disaster monitoring, and we discussed its mathematical foundations as well as the possible issues caused by this system, then gives few options to solve these issues.

In the chapter of functionalities, we illustrated a prototype GWAPs-based disaster monitoring human computation system for game players as well as stakeholders, and then described its necessary functions and interaction logic. On thoughts of system implementation, we

decided to implement this system on the web, probed the possible technology frameworks and pointed out the reason of our choices.

Afterward, in the chapter of design, we modeled the entire system theoretically in details that make sure it can run consistently. First of all, we defined a **Player Rating Model** for calculate a trust value of a player via artificial features, and then we put forward an algorithm that can be used in malicious user detection. As justification, we proved the correctness of this model. Meanwhile, as the data aggregation, we transferred the problem of calculating disaster level of regions into processing the expectation value of user tagging task inputs and proposed the **Disaster Evaluation Model**. In this model, we prevented the overabundance problem of potential useless tags from users by standing on the prespective of Bayesian. Surely, we addressed the solution of cold start of the human computation system. It is worth mentioning that the minimum initial trusted group under this scheme design only requires two persons theoretically.

Furthermore, as evaluation, we discussed theoretical evaluation criteria for this system, and then declared the challenges and corresponding solutions for facing issues like data security, information leakage, malicious detection as well as the lack of players. Undoubtedly, the current system design still contains defects. Thus, we presented three analysis and possible improvements for evaluation outdated, information loss and also gameplay playability.

For the future works, we will simply discuss the possible extension of our human computation system.

## 5.2 FUTURE WORKS

Our system was described in general. We collect human inputs by ROI tagging tasks, which means any other HC system that related to ROI tagging tasks can easily use this system back-end design. In addition, due to the fact that we do not have enough user inputs at present, we use a certainty algorithm instead of uncertainty probabilistic-based algorithm to detect malicious groups. Considering malicious detection is classification problem, which separate users into trusted groups and untrusted groups. One can apply any classification machine learning algorithms that are more suitable for the detection of malicious groups if our user input dataset is large enough.

Besides, as we mentioned before, sometimes, the game player may encounter a situation that there is no ROI in some pictures which contain only landscapes like: mountains, rivers and forests. In this case, with collaborative computing of image recognition technique, we can filter out those images from our image database previously so that we can collect more data from the game and make our image tagging game more efficiently.

## ACKNOWLEDGEMENTS

The authors would like to thank Prof. François Bry first for his great suggestions on information leak and loss problems of our disaster monitoring system; we also thank Yingding Wang for his helpful discussions on system functionalities design and the model rationalization evaluation; Finally, we also thank our schoolmate Huimin An for his inspiration of Bayesian perspective that helps us handling human inputs with new tags successfully.

The resources of this project such as paper  $\LaTeX$ code, mockup drafts as well as lab session beamer slides are open source on GitHub:

<https://github.com/changkun/hc-ss17-disaster-monitoring>.

## REFERENCES

- [BL01] Phillip Bonacich and Paulette Lloyd. Eigenvector-like measures of centrality for asymmetric relations. *Social networks*, 23(3):191–201, 2001.
- [BLZ13] Xiaojun Bi, Yang Li, and Shumin Zhai. FFitts law: modeling finger touch with fitts' law. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1363–1372. ACM, 2013.
- [GH16] Emma Graham-Harrison. 'I couldn't take anything except dignity': stories of the leaving of Aleppo, 2016.
- [HM82] James A Hanley and Barbara J McNeil. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, 143(1):29–36, 1982.
- [LJLB15] Mathieu Lafourcade, Alain Joubert, and Nathalie Le Brun. *Games with a Purpose (GWAPS)*. John Wiley & Sons, 2015.
- [PBMW99] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The PageRank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.
- [QB09] Alexander J Quinn and Benjamin B Bederson. A taxonomy of distributed human computation. *Human-Computer Interaction Lab Tech Report, University of Maryland*, 2009.
- [QB11] Alexander J Quinn and Benjamin B Bederson. Human computation: a survey and taxonomy of a growing field. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 1403–1412. ACM, 2011.
- [RV96] Raimundo Real and Juan M Vargas. The probabilistic basis of Jaccard's index of similarity. *Systematic biology*, 45(3):380–385, 1996.
- [Uni94] Unicef. *The state of the world's children*. 1998. Unicef, 1994.

- [VAD04] Luis Von Ahn and Laura Dabbish. Labeling images with a computer game. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 319–326. ACM, 2004.
- [WBBL13] Christoph Wieser, François Bry, Alexandre Bérard, and Richard Lagrange. ARTigo: building an artwork search engine with games and higher-order latent semantic analysis. In *First AAAI Conference on Human Computation and Crowdsourcing*, 2013.
- [Wik17a] Wikipedia. Jaccard index — Wikipedia, the free encyclopedia. <http://en.wikipedia.org/w/index.php?title=Jaccard%20index&oldid=788825557>, 2017. [Online; accessed 31-July-2017].
- [Wik17b] Wikipedia. Pearson correlation coefficient — Wikipedia, the free encyclopedia. <http://en.wikipedia.org/w/index.php?title=Pearson%20correlation%20coefficient&oldid=793244175>, 2017. [Online; accessed 31-July-2017].
- [Wik17c] Wikipedia. UNICEF — Wikipedia, the free encyclopedia. <http://en.wikipedia.org/w/index.php?title=UNICEF&oldid=792988632>, 2017. [Online; accessed 31-July-2017].