



Gonçalo Carrascal Tavares

Licenciado em Engenharia Informática

Learning Facial-Expression Models with Crowdsourcing

Dissertação para obtenção do Grau de Mestre em
Engenharia Informática

Orientador : Prof. Doutor João Miguel da Costa Magalhães, Prof.
Auxiliar, Universidade Nova de Lisboa

Júri:

Presidente: Prof. Doutor Pedro Manuel Corrêa Calvente Barahona

Arguente: Prof. Doutor Mário J. Gaspar da Silva

Vogal: Prof. Doutor João Miguel da Costa Magalhães



FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Dezembro, 2014

Learning Facial-Expression Models with Crowdsourcing

Copyright © Gonçalo Carrascal Tavares, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa

A Faculdade de Ciências e Tecnologia e a Universidade Nova de Lisboa têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objectivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

In loving memory of my grandmother

Acknowledgements

First and foremost, I would like to express my genuine gratitude to my advisor Professor João Magalhães, not only for all the support, patience, guidance and knowledge, but also for developing my engineering skills, which will be helpful in my professional and personal life. Additionally, I want to acknowledge my lab colleagues, André Mourão, Flávio Martins and Filipa Peleja for all the advises and knowledge.

Second, I would like to thank Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa to provide all the resources needed to accomplish my professional goals, with a special thanks to my department, Departamento de Informática. Moreover, I would like to thank the Fundação para a Ciência e a Tecnologia for providing my scholarship.

Third, I want to give a huge thanks to all my warrior friends: André Grossinho, Catarina Gralha, Diogo Cordeiro, Gabriel Marcondes, Hélder Gregório, Hélder Marques, João Claro, João Espada, João Ferreira, who also fought their own battles. Writing a thesis becomes much easier when we are all together. In special, I want to thank all *The Mumbling* community for all the hours of fun spent together, which helped in relieving stress. It redefined the concept of *all night long*.

To all my family who is always there when I need the most. To my mother and father who made me in the man that I am today and for all the emotional and financial support. The most deepest thanks to my beloved girlfriend, Rita Belo, for these wonderful eight years where I not only learned the meaning of the word *love* but also, and most importantly, the meaning of friendship, support and encouragement.

I would like to send a special thanks to Goku, my best childhood friend, with whom I shared great moments defeating powerful enemies, while saving the planet Earth. Moreover, he taught me how to perform a *Universal Spirit Bomb* to absorb all my friends strength and use it to finish this thesis.

Abstract

The computational power is increasing day by day. Despite that, there are some tasks that are still difficult or even impossible for a computer to perform. For example, while identifying a facial expression is easy for a human, for a computer it is an area in development. To tackle this and similar issues, crowdsourcing has grown as a way to use human computation in a large scale.

Crowdsourcing is a novel approach to collect labels in a fast and cheap manner, by *sourcing* the labels from the *crowds*. However, these labels lack reliability since annotators are not guaranteed to have any expertise in the field. This fact has led to a new research area where we must create or adapt annotation models to handle these weakly-labeled data. Current techniques explore the annotators' expertise and the task difficulty as variables that influences labels' correction. Other specific aspects are also considered by noisy-labels analysis techniques.

The main contribution of this thesis is the process to collect reliable crowdsourcing labels for a facial expressions dataset. This process consists in two steps: first, we design our crowdsourcing tasks to collect annotators labels; next, we infer the *true* label from the collected labels by applying state-of-art crowdsourcing algorithms. At the same time, a facial expression dataset is created, containing 40.000 images and respective labels. At the end, we publish the resulting dataset.

Keywords: Crowdsourcing, facial expressions, machine learning.

Resumo

O poder computacional tem crescido de dia para dia. Apesar disso, continuam a existir tarefas que são bastante difíceis ou mesmo impossíveis para um computador realizar. Uma destas tarefas é identificar uma expressão facial. Esta tarefa é fácil para um humano, mas para um computador é uma área em desenvolvimento. Para atacar problemas deste tipo, *crowdsourcing* cresceu como uma forma de usar *human computation* em larga escala.

Crowdsourcing é uma nova abordagem para recolher anotações de uma maneira fácil e económica. No entanto, estas anotações carecem de credibilidade. Isto possibilita uma nova área de pesquisa onde teremos de criar ou adaptar modelos de anotação para suportarem estas anotações fracas. As técnicas actuais, exploram a especialização de cada um dos anotadores e a dificuldade de cada tarefa como variáveis que controlam a correcção de uma anotação. Outros aspectos específicos são também considerados pelas técnicas de análise de anotações com ruído.

A principal contribuição desta tese é o processo de recolher anotações fiáveis através de crowdsourcing para um conjunto de dados de expressões faciais. Este processo é composto por duas fases: Primeiro é necessário recolher as anotações por crowdsourcing, que passa por definir quais os parâmetros da nossa tarefa. Após recolhidas várias anotações por imagem, é necessário inferir a verdadeira anotação entre estas. Para isso serão estudados os modelos de crowdsourcing mais conhecidos. Ao longo deste processo será criado um conjunto de dados com mais de 40.000 expressões faciais.

Palavras-chave: Crowdsourcing, expressões faciais, aprendizagem de máquina.

Contents

1	Introduction	1
1.1	Human-based Computation	1
1.2	Motivation	3
1.3	Affective-interaction Data	3
1.3.1	Data Acquisition Setting	4
1.4	Organization	5
1.5	Problem Formalization	5
2	Background and Related Work	7
2.1	Human Computation	7
2.2	Crowdsourcing Research Fields	8
2.2.1	Interaction Strategies	9
2.2.2	Performance	11
2.2.3	Human in the Loop	11
2.2.4	Dataset	12
2.2.5	Types of Judgements	13
2.3	Annotation Models for Weakly-labeled Data	14
2.4	Facial Expressions Datasets	16
2.5	Summary	18
3	Crowdsourcing	
	Facial Expressions Labels	19
3.1	Building a Real-World Facial Expressions Dataset	19
3.2	Optimizing Intra-Worker Agreement	20
3.3	Crowdsourcing Task Design	21
3.3.1	Answers domain	21
3.3.2	Selecting a Pool of Workers	22
3.3.3	Job Attributes	23
3.4	Tuning Jobs	23

3.4.1	Setup	24
3.4.2	Number of Judgements per Image	24
3.4.3	Number of Judgements per Worker	26
3.4.4	Worker’s Payment vs Geographic Location	27
3.5	Full Jobs: Results and Discussion	28
3.6	Summary	32
4	Benchmarking Weak-labels Combination Strategies	35
4.1	Crowdsourcing Methods	35
4.1.1	Majority (MV) and ZenCrowd (ZC)	35
4.1.2	Dawid and Skene (DS)	36
4.1.3	GLAD	37
4.1.4	CUBAM	37
4.1.5	Raykar (RY)	38
4.2	Experimental Setup	38
4.3	Synthetic Experiment: Modelling Workers Expertise	40
4.3.1	Gaussian	41
4.3.2	Inverse Gaussian	42
4.3.3	Logistic	43
4.3.4	Gaussian Translated	43
4.3.5	Discussion	44
4.4	Real Data Experiment	45
4.4.1	Dataset	45
4.4.2	Results	46
4.5	Hybrid Experiment	46
4.5.1	Random Workers	47
4.5.2	Adversarial Workers	49
4.6	Summary	50
5	Learning Classifiers with Weak-labels	53
5.1	Methodology	53
5.1.1	k -Nearest Neighbors	54
5.1.2	Kernel density estimation	54
5.2	Datasets	55
5.2.1	Cohn-Kanade	56
5.2.2	Novaemotions	56
5.2.3	Crowdsourcing labels	56
5.3	Classifying Facial Expressions with Weak Labels	57
5.3.1	Training and Test Data	57
5.3.2	Results	58
5.4	Summary	60

CONTENTS

xv

6 Conclusion	61
6.1 Future work	63

List of Figures

1.1	The game interface.	4
1.2	Example of Novaemotions dataset.	4
1.3	Example of CK dataset.	5
2.1	A diagram of crowdsourcing research fields.	9
2.2	An example of drawing from The Sheep Market.	11
2.3	A subtree of ImageNet database.	12
3.1	Example faces from the dataset.	20
3.2	The gamification and crowdsourcing processes to generate the facial expressions dataset.	21
3.3	Worker interface.	22
3.4	Average of image's agreement over number of judgements. The blue line is a random sequence of judgements per image, the red line is the worst sequence, and lastly, the green is the best sequence.	25
3.5	The blue line is the average of agreement for the n th judgement of each worker. Each red dot is the agreement of one worker for the n th judgement. The area around the blue line is standard deviation. On the left side, is presented the analysis of the first job where workers. On the right side. . .	26
3.6	Worker's location	27
3.7	Example of two images whose the label is <i>Not a face</i>	28
3.8	Workers agreement with the selected label, sorted by agreement	31
4.1	The graphical representation of GLAD model.	37
4.2	A simplified version of the graphical representation of CUBAM model. . .	37
4.3	Accuracy of crowdsourcing methods using a constant distribution of worker's expertise	41
4.4	Accuracy of crowdsourcing methods using a normal distribution of worker's expertise	42

4.5	Accuracy of crowdsourcing methods using a inverse Gaussian distribution of worker's expertise	43
4.6	Accuracy of crowdsourcing methods using a logistic distribution of worker's expertise	43
4.7	Accuracy of crowdsourcing methods using a Gaussian distribution translated of worker's expertise	44
4.8	Species that each annotator has to identify	45
4.9	Accuracy of running crowdsourcing methods using bluebirds dataset. . .	47
4.10	Accuracy of running crowdsourcing methods using bluebirds dataset with random workers.	48
4.11	Accuracy of running crowdsourcing methods using bluebirds dataset with adversarial workers.	50
5.1	Example of CK+ dataset.	56
5.2	Example of Novaemotions dataset.	56

List of Tables

2.1	A comparison between the most popular facial expressions datasets [59]. .	17
3.1	Job's parameters.	23
3.2	Number of votes of each facial expression for each job.	29
3.3	Confusion matrix for each facial expression	29
3.4	Workers' votes for facial expressions wiht lowest and highest agreement. .	30
4.1	Average of relative accuracies when comparing against the majority voting.	45
5.1	Precision of each model labels against known groundtruth	57
5.2	Accuracies of training various classifiers (k -NN, weighted k -NN and KDE) using crowdsourcing labels from different methods.	59



Introduction

The scientific interest in computer vision methods has increased over the past years. However, the intrinsic human capabilities at performing semantic analysis are far more advanced than computers. For this reason, researchers have developed ways to integrate humans as a source of computation. This gave way for crowdsourcing to emerge. Crowdsourcing provides a fast and reliable way to request workers. One such task could be identifying the facial expression present in a given image. While this is a trivial matter for a human, the same is not true for computers. Crowdsourcing was the answer to replace computers in this kind of tasks. It can be used within the research community to build entire or partial datasets. However, to take full advantage of crowdsourcing is not a trivial task, since it involves working with anonymous people from around the world. This means there is no access to a given person's work environment nor any assessment concerning their competence. Thus, several methods have been proposed in recent years to improve the quality of crowdsourcing data.

The main focus of this thesis is to figure the best approach to build a facial expression dataset entirely through crowdsourcing. First, we will study the best way to collect the crowdsourcing data. Second, we will study the crowdsourcing methods to maximize the data quality. Finally, we will test how such data can be used to train a classifier as an alternative to the ground truth given by experts.

1.1 Human-based Computation

Despite the remarkable advancements in computing capabilities and efficiency, computers are still outperformed by humans at certain tasks. The power of perception and interpretation are functions that us, humans, perform naturally with less or non effort in

our day-to-day lives. While computers are capable of executing millions of instructions per millisecond, they cannot yet perform certain tasks at the same level as a human, e.g. interpreting an image's content, since computers do not have this innate ability. Human based computation, the act of taking advantage of these intrinsic human abilities as a computation source, is a science dedicated to addressing this issue. A human-based computation (HBC) algorithm divides a given problem in small parts – micro tasks – and then assigns it to someone – the worker – who's responsible for solving it. The answer from each worker is then used. The use of HBC has become commonplace in research environments where researchers appeal to volunteers or hire people to work as a source of computation. Despite the success of HBC in research environment, the ease of some tasks associated with the emergence of internet in our homes made possible to expand the concept of HBC to a larger scale. This was the origin of crowdsourcing.

Crowdsourcing is the process of collecting data from a network of people where an unknown person around the world can volunteer or be hired in place of a traditional worker: a natural expansion of HBC. Since some problems rely more on human interpretation, this means that most people are skilled enough to become workers and this outlines the utility of crowdsourcing. The advantages of crowdsourcing go beyond worker's skill. Another advantage is that it is less expensive to hire a worker than an expert worker: a qualified person to perform a certain task. This is crucial when working with problems involving high amounts of data, which will ideally require contracting several expert workers that will become quickly unaffordable. Another aspect is the time required to process the data: by hiring a virtually unlimited number of workers, researchers can reduce the waiting time of a job.

Crowdsourcing give us enormous advantages. However, there are some aspects that must be considered when using crowdsourcing: given that a worker is anonymous, there are no guarantees that she is qualified to perform a specific task. However, there are some aspects that must be considered when using crowdsourcing: first, it is strongly dependent of the nature of the task; second, given that a worker is anonymous, there are no guarantees that she is qualified to perform a specific task and, even when the task is performed by a qualified worker there are no guarantees that it will be executed correctly once some aspects, such as distraction, fatigue or boredom, are not being accounted. Some of these effects cannot be controlled, and Chapter 3 will present a study on how to best attenuate this issue. One simple solution is to hire more than one worker for the same task. However, we must handle the case where not all workers answer the same. A naive approach to infer the *true* answer of a task is to choose by a majority voting approach where the answer with more votes is set as the *true* answer. However, we will study other approaches that outperform the majority voting. These approaches consider other factors such as task difficulty and worker expertise, which allows to predict with more precision the *true* label.

1.2 Motivation

The emergence of crowdsourcing opens the doors to develop many research fields. One of these fields is machine learning. It allows to build larger and richer datasets that were impossible until now and to have more and diverse training sets to improve computer vision algorithms. In facial expression recognition problems, we usually resort to actors or psychologists to perform a given facial expression. These expressions are not entirely genuine, since the emotions behind them are being forced. Moreover, they are taken in an environment where the pose, lighting and other aspects are controlled. Does it make sense to *teach* computers with samples that do not reflect what happens in the outside world? In order to create a more realistic dataset, a game has been developed [1] to capture the players' facial expressions while interacting with it. Although not entirely genuine, they were captured in a realistic scenario in which players were aware their facial expression were in full control of the gameplay. Throughout several game sessions, a large set of unlabelled interaction images was collected. Ideally, we must hire a expert to label this dataset. However, this process quickly becomes unaffordable for a large number of images. We believe that crowdsourcing can help us collect this type of labels in a faster, cheapest and reliable manner. Therefore, the main goal of this thesis is:

**research the viability of crowdsourcing to create facial expressions
ground truth.**

1.3 Affective-interaction Data

The process of collecting the affective interaction data consists in a two-player game where the aim is to perform a number of facial expressions to match several emotions displayed, within a certain time limit, in a series of rounds. Players play simultaneously and each player's facial expression is scored based on proximity to the emotion asked for. The player who achieves the highest score wins that round, and the player with the highest number of won rounds wins the game. This game is described in greater detail in [1].

Figure 1.1 shows the game's main interface. The colored bars represent the scores (top: previous image score, bottom: best score of the round); the numbers in the center represent the global scores; the half circle is the round timer; the image label in the middle is the emotion the players are being asked to mimic and the image attempts to help evoking that emotion; the faces on both sides are the players and the label represents the last expression recognized. Refer to [2] for details on the facial expressions analysis algorithm.

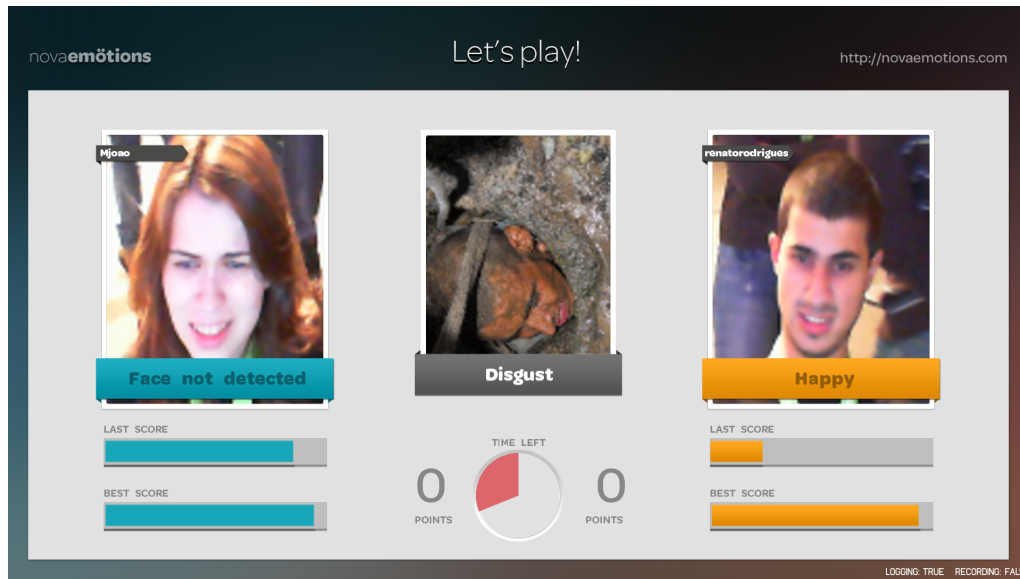


Figure 1.1: The game interface.

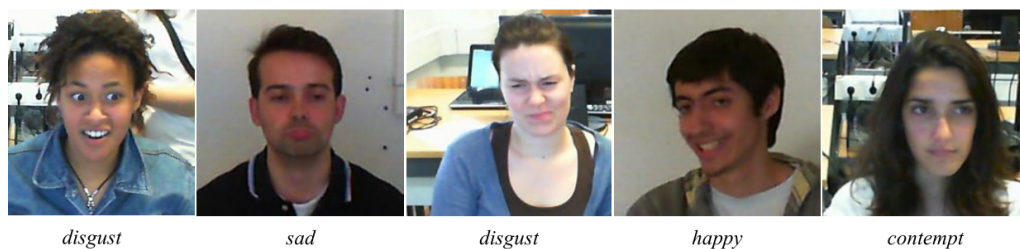


Figure 1.2: Example of Novaemotions dataset.

1.3.1 Data Acquisition Setting

A total of 42,911 facial expressions were captured during the game sessions. These expressions were captured in a novel and realistic setting: humans competing in a game where their facial expressions have an impact on the outcome. Some examples are visible in Figure 1.2. These images offer a novel view of facial expression datasets: players were competing using their own facial expressions as an interaction mechanism, instead of performing a well predefined prototype expression.

This dataset is also unique in the following senses: users faces are not in fixed positions (about 50% of the face images are not front facing and they are at different heights). Existing facial expressions datasets like CK+ [3] or the BU-4DFE [4] datasets were captured in controlled environments. Moreover, facial expressions of CK+ dataset were captured by people trained to perform a prototype expression. Some examples are displayed in Figure 1.3.

Our approach was different: first, the volunteers were asked to perform an expression in a social gaming environment with varying lighting, background and position; second,



Figure 1.3: Example of CK dataset.

a pure affective-interaction setting where the computer is controlled by the players facial expression; and lastly, each captured image contains the information regarding the expected expression and the one detected by the game algorithm.

1.4 Organization

This thesis is organized as:

- **Chapter 3: Crowdsourcing Facial expressions Labels:** A crowdsourcing process must be used to label each image with a facial expression [5]. Given the large number of images collected, it was not possible to have them labelled by an expert. Thus, for that purpose, a crowdsourcing service must be used. This subject will be explored in Chapter 3.
- **Chapter 4: Benchmarking Weak-labels Combination Strategies:** Once the labels from the different workers are obtained, the labels concerning the same image need to be merged. Different strategies exist to achieve this goal. Therefore, several popular techniques will be compared and analysed in Chapter 4.
- **Chapter 5: Learning Classifiers with Weak-labels:** The final purpose of this research is to infer if labelling through crowdsourcing is a good and less expensive alternative to the ground truth provided by experts. Chapter 5 details the training of three classifiers with the label set obtained through crowdsourcing and compare it against a ground truth of the same set.

1.5 Problem Formalization

In this thesis the following notation will be used: A set of N images indexed by $\mathcal{I} = \{1, \dots, N\}$ where each image i has one true label: the ground truth. This is denoted by $z_i \in \mathcal{Z} = \{z_1, \dots, z_N\}$ that belong to a set of possible labels or classes \mathcal{C} . A set of M workers indexed by $\mathcal{J} = \{1, \dots, M\}$ will label a subset of images $\mathcal{L}^j = \{l_{ij}\}_{i \in \mathcal{I}_j}$, where $\mathcal{I}_j \subseteq \mathcal{I}$ is the set of all labels produced by the annotator j . Similarly, all labels of an image i are denoted as $\mathcal{L}_i = \{l_{ij}\}_{j \in \mathcal{J}_i}$, where $\mathcal{J}_i \subseteq \mathcal{J}$. Also, the visual features of an image i is given by x_i .

An annotation model for weakly-labeled data or crowdsourcing method, produces a set of estimated labels $\mathcal{Y} = \{y_0, \dots, y_N\}$ where $y_i \in \mathcal{Y}$ is the estimated label of image i .

The objective of this thesis is to figure the best approach to collect a set of reliable labels \mathcal{L} and estimate a set of labels \mathcal{Y} to maximize the likelihood between these and \mathcal{Z} .



Background and Related Work

2.1 Human Computation

As previously mentioned, despite the advances in the computer processing, certain tasks, such as understanding human languages, or semantically analysing an image, are still better performed by humans. Therefore, researchers have been looking for ways to use humans as a computation source: this is known as human computation [6]. Yuen et al. [7] phased human computation systems as follows:

Alternative to machines In the beginning, human computation was used as an alternative for tasks that are difficult to perform by computers, such as reasoning tasks. Moreover, the majority of these tasks are naturally performed by humans, for example, distinguishing an image's components.

Crowdsourcing A second generation of human computation was expanded to use multiple internet users to solve larger and/or harder problems. For instance, Wikipedia is a distributed human computation system where users around the world contribute with their knowledge to build an online encyclopedia.

Gamification The need to give more incentives to make users take part of a human computation task allied to the large amount of online gamers. Games are seen as a good way to entertain users while they are producing human computation data.

2.2 Crowdsourcing Research Fields

The expansion of internet has opened the doors for crowdsourcing to emerge. The term crowdsourcing was coined in 2006 by Jeff Howe [8]. In his website [9], Howe defines crowdsourcing as follows:

"Crowdsourcing is the act of taking a job traditionally performed by a designated agent (usually an employee) and outsourcing it to an undefined, generally large group of people in the form of an open call."

Crowdsourcing can be seen as a distributed human computation model in a professional environment, where employees are unknown users around the world. Quinn and Bederson distinguishes human computation from crowdsourcing: *"Whereas human computation replaces computers with humans, crowdsourcing replaces traditional human workers with members of the public."* [10]. There are many advantages in using crowdsourcing. It has been shown [11] that crowdsourcing's results are reliable and a good alternative to experts' results, such as natural language annotation, since only a small number of crowdsourcing workers are needed to equal the performance of an expert annotator [12]. Therefore, producing crowdsourcing data is much cheaper than hiring experts. However, crowdsourcing workers are not machines nor experts. Thus, incentives are needed to ensure their commitment to produce reliable results. The main incentive in crowdsourcing is money: a worker receives a small payment for each completed task. However, money is not the only incentive because workers can be motivated by the idea of contributing to a "greater good" [13]. This "greater good" can be astronomy research [14], or contributing to create synthetic RNA designs [15]. Another type of approach is designing the crowdsourcing application as a game. These games, where the final goal is to collect some crowdsourcing data, are known in literature as Games With a Purpose (GWAPs) [16]. One of the most known GWAP is the ESP game [17], where the final goal is to collect an image's labels.

In the literature, crowdsourcing is usually confused with crowdsourcing sites. Crowdsourcing sites, such as Amazon's Mechanical Turk (MTurk) [18] and Crowdflower [19], are sites with the purpose of collecting crowdsourcing data. These sites have two groups of people, namely the requesters and the workers, where requesters hire workers to perform a given set of tasks. However, there are other crowdsourcing approaches, such as Wikipedia, where users share information between themselves without involvement of payment.

Crowdsourcing research fields go beyond collecting crowdsourcing data. Yuen, King, and Leung [20] categorized these in four types: application, algorithm, performance and dataset. However, the expansion of crowdsourcing in later years reshaped some research fields and created new ones. Figure 2.1 outlines a more up-to-date categorization of crowdsourcing research fields. Therefore, we will present an overview of the most relevant ones.

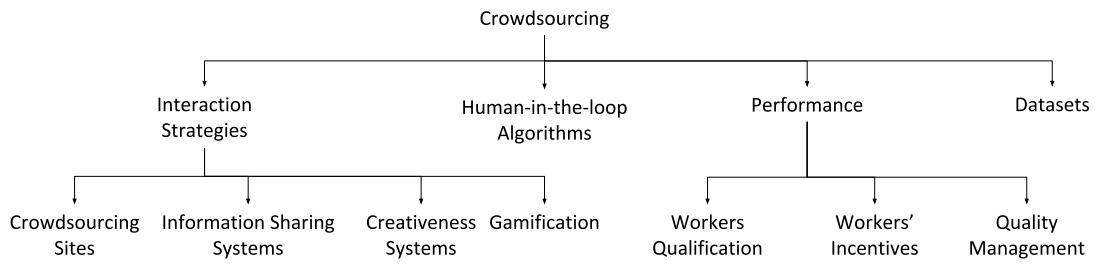


Figure 2.1: A diagram of crowdsourcing research fields.

2.2.1 Interaction Strategies

Crowdsourcing is an approach to collect knowledge. However, there are many methods to extract this human knowledge. We need to choose carefully the type of application that we need to build. The most common method is to use a crowdsourcing site to submit a job (set of tasks). Then, the crowdsourcing workers undertake one or more tasks and, after completing each task, they receive a reward. Other methods to collect human knowledge include building a web-based game or a web-platform to share information. In the second case, the reward is not monetary.

2.2.1.1 Crowdsourcing Sites

One of the uses of crowdsourcing is the possibility of collecting annotations. Crowdsourcing sites like Amazon mTurk and Crowdfunder made this process automatic in many ways. For instance, crowdsourcing sites give us a platform to create our tasks and automatically spread these tasks to crowdworkers. The way of collecting crowdsourcing data in these sites is similar to a questionnaire. The template for each task is a voting system (single or multiple choice), a text box (single or multiple lines), or a rating system. The voting system is the most used template in crowdsourcing sites, mainly to ease the posterior analysis of data and to limit the answer domain per task. In this system, a crowdsourcing worker must choose an answer from a set of possible choices. Despite collecting annotations, there are other purposes for crowdsourcing sites, such as: extracting opinions, collecting common-sense information, collecting relevance judgements, or even sentiment analysis. However, there are other ways to collect annotations through crowdsourcing. Julia Moehrmann [21] proposed an annotation tool to label large datasets by using a self-organized map (SOM) to cluster image data. With this approach, the user can label many similar images at once. Another approach tool is presented in [22] where users have to select a person's silhouette using four different protocols: two different coarse object segmentation protocols, polygonal labeling, and 14-point human landmark labelling.

2.2.1.2 Information Sharing Systems

There are some cases where the user has a more complex role in the crowdsourcing task. One of these cases are the information sharing systems, where information is shared between different users. Thus, there is no associated payment. An user can be seen as a worker and a requester: he shares information voluntarily and uses the available information freely. Application such as Wikipedia and Yahoo!Answers are real examples where crowdsourcing is used to share information between internet users. A novel approach in this area is the smartphone application StreetBump [23]. The concept behind this app is to collect road conditions such as holes or bumps. This information is not only useful for helping improving road conditions but also for other users who want to avoid roads that are in bad conditions.

2.2.1.3 Gamification

Gamification is the process of collecting data from a group of users through a game [24, 25]. The main purpose of using a game as a crowdsourcing application is the inherent enjoyment of games, which works as an extra incentive to users. Some data or knowledge are hard to collect due to different reasons, such as the natural repetitiveness or the long durations of some tasks, which causes the users to lose interest. Games are the application type that keeps the users engaged and motivated. The benefits of gamification have been explored in recent years by the community to collect data: unlike laboratory tasks, games provide a fun and engaging environment for the user and, consequently, the produced results are more reliable and genuine. The ESP game [17] is an example of a human computation game. The game consists in providing images for two different players to label. Once both players write the same label, not necessary at the same time, that label becomes that image's label. Other good examples of human computation games are ZoneTag [26] and reCAPTCHA [27]. The EteRNA [28] [15] is a browser game, where the goal is to create sequences of ribonucleic acid (RNA), which is an essential molecule for controlling several cellular processes. Thus, in this game there are some rules to create these RNA sequences which mimic the chemical interactions that exist in nature. Therefore, the ultimate aim of this game is to create a large-scale database of synthetic three-dimensional RNA sequences.

2.2.1.4 Crowd Creativeness

Lastly, we have the tasks that need more than human knowledge. Tasks like drawing or writing a book requires creativity which is perhaps the hardest human skill to simulate. The sheep market [29] is an interesting experience to test the crowdsourcing user's creativity. This experiment consisted in paying 0.02\$ to a crowdsourcing worker to draw a sheep facing left. This resulted in a database with 10.000 drawings, and some of these drawings are presented in Figure 2.2.



Figure 2.2: An example of drawing from The Sheep Market.

2.2.2 Performance

The performance aspect of crowdsourcing systems is very important, since it defines how reliable the crowdsourcing output will be. We are used to associate performance to a computer. In this case, we are interested in optimize human performance. Therefore, the attributes that we must be aware of are different from computers' attributes, such as the worker eligibility. For example, in the context of this thesis, one must be aware that facial expressions are not universal: different cultures have different ways of expressing emotions [30]. Another important attribute is the price to pay for each task. Mason and Watts [31] presented a study where they compared the relationship between financial incentives and performance. They concluded that increasing the payments increased the quantity of work performed by each user, but not its quality. The time to complete a crowdsourcing task has also been researched. Bernstein et al. [32] showed that is possible to recruit a worker in two seconds and complete all the crowdsourcing process in ten.

2.2.3 Human in the Loop

Sometimes crowdsourcing is a mean to an end, not an end in itself. In other words, crowdsourcing can be part of some system and not be the system itself. Crowdsourcing is widely used in computer algorithms as a way to perform or validate some computer tasks. The systems that use crowdsourcing workers are usually called human-in-the-loop systems. The CrowdDB system [33] aims at extending the SQL language by using crowdsourcing to solve some problems in the database, such as (1) unknown or incomplete data and (2) subjective comparisons. When the system detects such faults, it automatically generates an user interface in Mechanical Turk and the task to request a worker to solve the problem. After the crowdsourcing process finishes, the collected information is inserted in the database.

Crowdsourcing can also be useful to improve results in the field of image search. Human skills are not just a good alternative to computers algorithms but they are also a good way to validate them. One example of this concept is the image search system CrowdSearch, where it combines automated image search, by taking a picture with a

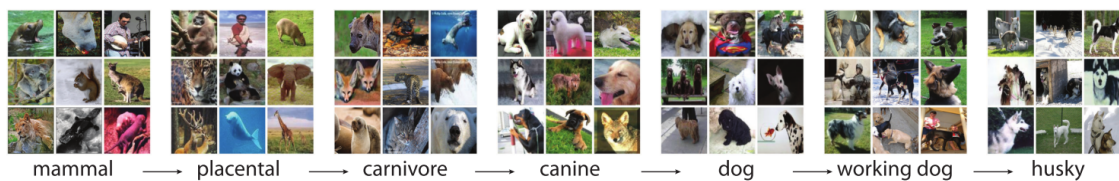


Figure 2.3: A subtree of ImageNet database.

smartphone, with real-time human validation where Then, image features are extracted and uploaded to a back-end server where an automated image search engine is executed to find a set of similar images (candidate images). The resulting set of images is given to crowdsourcing workers to validate them: workers specify if the candidate images contain the same object as the picture or not.

2.2.4 Dataset

As shown in the previous section, a lot of research has been made to improve crowdsourcing algorithms, we need datasets to validate these algorithms. In fact, several crowdsourcing datasets have been published, which also enable the improvement of the state-of-the-art algorithms. Dataset ground-truth can be divided in three categories: (1) binary; (2) multiple-level; (3) ranked. These categories are determined by the type of the judgements collected (see Section 2.2.5).

There are many crowdsourcing datasets available for research. One of the most popular is the dataset collected with the ESP Game [17] which consists of 100,000 images with the respective labels. This proved the advantages of using crowdsourcing as a way to create a large dataset at a low cost. Other good example is the Galaxy Zoo [14], and their later extended version - the Galaxy Zoo 2 [34], which is a web-based project where everyone can participate by helping to catalogue a large number of galaxies. This project resulted in a dataset [35] [36] with nearly one million galaxies annotated by 100,000 participants, corresponding to a total total of 40 million judgements, which represents an enourms advance in the astronomical field. On the other hand, crowdsourcing can be also used to validate computer results. This idea was used by Deng et al. [37] to create the ImageNet database. In fact, they build an image database based on the WordNets hierarchical dataset [38] structure. This process consisted on collecting 500-1000 candidate images from search engines for each node of WordNets, which are called synsets Deng et al. All synsets are hierarchical connected through a hyponymy relation (i.e. the node "dog" is a child of the "canine"). In Figure 2.3. we also represent one example of a subtree of this database. Nowadays, this database includes 21,841 synsets with an overall of 14,197,122 images (an average of 650 images per synset). Finally, to validate the images, the authors resorted to a crowdsourcing task in the Amazon Mturk, where they presented to each worker a candidate image and its respective synset. Thus, workers had

to specify if the presented image contained one possible object of the presented synset.

2.2.5 Types of Judgements

Depending on the information domain, different definitions of relevance are more adequate than others. Three types of relevance judgments are easily identified in the literature:

- **Binary relevance:** under this model a document is either relevant or not. It makes the simple assumption that *all* relevant items contain the same amount of information value.
- **Multi-level relevance:** one knows that documents contain information with different importance for the same query, thus, a discrete model of relevance (e.g., relevant, highly-relevant, not-relevant) enables systems to rank documents by their relative importance.
- **Ranked relevance:** when documents are ordered according to a particular notion of similarity.

The binary relevance model is the most common practice in HCI and AI systems. These systems are tuned with a set of judgements that reflect the majority of experts' judgements. The multi-level relevance provides the annotator with more expressive power than with binary relevance - e.g. workers feel more comfortable with three or four levels of relevance-intensity instead of only true/false. The relevance judgements of the ranked relevance model are actually a rank of documents that exemplify the human perception of a particular type of similarity, e.g., texture, colour. In practice, for the task at hand, only the binary or the multi-level judgements are viable.

2.2.5.1 Assessors Agreement

The judgements quality of crowdsourcing jobs has been the matter of much research, [11]. Traditionally, expert annotations are obtained through processes that eliminate problems of inconsistency and bias. Volkmer, Thom, and Tahaghoghi [39] followed the following rules to improve judgements' quality: (1) assessors annotated a sub-set of the documents with a sub-set of the labels (this avoids the bias caused by having the same person annotating all data with the same concept); (2) all documents must receive a relevance judgement from all annotators (this eliminates the problem of incomplete relevance judgements but increases inconsistency); (3) documents and labels were assigned to annotators so that some documents received more than one relevance judgement for the same label, thereby eliminating the inconsistency problem if a voting scheme is used to decide between relevant and non-relevant. This annotation study was a quite formal and expensive process. Nowak and R uger [11] compared the judgements quality of expert to that

of individual workers. They confirmed through several statistical measures that the aggregated results of the non-experts (crowdsourcing workers) are comparable to the ones created by experts.

2.3 Annotation Models for Weakly-labeled Data

Labels produced by non-expert annotators lack credibility. To tackle this problem, many methods have been proposed to infer true labels from weakly-labels. The most common method is the "majority vote", where the estimated label is the label with more votes. This approach works well in our society when we have to elect someone: for example, a president, since every citizen's opinion have the same weight in this process. However, the annotation process where many annotators contribute with their own labels does not need to function in the same manner, since every worker has his own area of expertise and his interpretation may be different from the others. For this reason, many models that outperform the majority vote have been proposed. In this section we will briefly describe some annotation models. However, in Chapter 4 we will present a more detailed explanation of the most known annotation models: Dawid and Skene (DS) [40], GLAD [41], CUBAM [42], Raykar [43], Zen [44] and Majority vote.

The majority of these models are based on the Expectation Maximization (EM) algorithm [45]. Dawid and Skene [40] were the first to use the EM to infer the experts' bias and also the unknown label. However, more models were researched after the expansion of crowdsourcing.

Later in 1995, Smyth et al. [46] proposed a model based on EM algorithm to infer the true label from a set of expert annotators. They produced an experiment using 4 expert annotators which had to label a ROI's (Region of Interest) for satellite images of Venus. Annotators had to choose if the ROI contained a vulcano or not. To do so, annotators could choose between 5 confidence levels. Later, due to the simplicity of collecting labels from non-expert annotators, the necessity to clean noisy labels from datasets started to grow.

In 2009, many models emerged to tackle the problem of weakly-label data. Donmez, Carbonell, and Schneider [47] were the first to propose a method to infer each annotator's expertise and choose the best ones for active learning. This model is based on a Interval Estimation (IE) learning algorithm, which estimates a confidence interval of an expected response, given an action, and then returns the highest confidence interval. With this approach, the authors use the IE assuming that the action is: choose an annotator from an oracle to label a dataset unit. This strategy allows us to estimate the most competent annotators to label a given dataset unit.

Later, Whitehill et al. presented GLAD, a probabilistic model to infer the probability of an attribute being present in an image. Unlike majority vote, GLAD estimates not only the true label but also the annotators accuracy and the image difficulty. To estimate these parameters, GLAD uses Expectation-Maximization approach (EM). This model is

frequently used in literature.

Peter Welinder et al. proposed the CUBAM model [42] which is a generalization of the GLAD model. In this model, the image difficulty is represented by a high-dimensional concept. Instead of being parametrized by a value representing the image difficulty, an image is parametrized by a vector of task-specific measurements x_i that are available to the visual system of an ideal annotator. A way of thinking about it is to consider the vector to be a representation of visual features. Different annotators have different interpretations of the same image, therefore, each annotator will perceive a corrupted version of x_i , due to the noise n_{ij} caused by his interpretation $y_i = x_i + n_{ij}$. Each annotator is parametrized with a vector w_j representing the annotator's expertise among all components of the image. With this approach, each annotator is not classified as being good or bad. Instead, this model finds the "areas of strength" of each annotator.

Simultaneously, the same author proposed an online crowdsourcing algorithm [48] based on EM algorithm for requesting only labels for instances whose true label is uncertain. Additionally, they infer who are the best annotators, prioritize these, and block the possibly noisy annotators. They formalize the expertise of each annotator for binary, multi-valued and continuous annotations.

Another model based on EM algorithm was proposed by Raykar et al. [43], where they estimate the true label while modelling a classifier. This model assumes that each annotator has two attributes: sensitivity and specificity. If we consider a binary model where labels can assume values 0 or 1, these attributes can be interpreted as the expertise of the annotator in identifying the label 1 and label 0, respectively. Then they use the EM algorithm to iteratively compute the maximum-likelihood and the optimal values for the sensitivity and specificity that maximizes this likelihood. Chittaranjan, Aran, and Gatica-Perez [49] proposed a similar solution in the context of detecting psychological traits. However, these models assume that the expertise of an annotator is not dependent of the instance to label.

Tang and Lease [50] proposed a semi-supervised naive bayes approach. This model assumes that the input data has both unlabeled data and data labeled by experts. Initially, the unlabeled data is labeled by the majority vote of all worker's votes. Then, they use a EM approach to iteratively find two parameters: the set of class priors and the probability of a given worker classifying an example of a specific class given the true label. Based on this parameters, they re-estimate the labels for the initial set of unlabeled data. This process is repeated until convergence.

Kamar, Hacker, and Horvitz [51] proposed a system called CrowdSynth, which uses machine vision, machine learning and decision-make to predict both the correct label and the necessity to hire more workers to achieve a good agreement. They created a model using a Naive Bayes approach to predict the correct label of a given task. At the end of each iteration, the system runs a second model that predicts the state of the system that would come with the hiring of an addition worker and decides if is needed to hire more workers or if the execution can be terminated. This system was modelled with a Markov

Decision Process (MDP) approach.

Mcduff et al. [52] collected and analysed crowdsourcing data of dynamic, natural and spontaneous facial responses while viewers were watching online media. The authors believe that facial responses are a good way to measure user's engagement with content. Such information is ultimately for content creators, marketers and advertisers.

2.4 Facial Expressions Datasets

Humans are able to recognize different facial expressions and infer what emotion that expression conveys. Ekman [53] defined a total of six basic universal emotion expressions: *Happiness*, *Sadness*, *Surprise*, *Fear*, *Anger* and *Disgust*. *Neutral*, a state of no visible expression, and *Contempt*, a mixture of *Anger* and *Disgust*, are also part of Ekman's suit of facial expressions of emotions. These are the expressions we have chosen in our work. However, changes in facial expression can be more subtle, like moving the outer section of the brows or depressing the corners of the lips. The expressions described above can be defined as a set of Action Units (AUs). An AU is an action performed by one or more muscles of the face that humans are able to distinguish. A full description is available at Tian, Kanade, and Cohn [54]. Even using AUs, some expressions are similar: *Fear* is composed by the same AUs as *surprise* plus other 3 AUs [55].

Due to the difficulty of labeling facial expressions by a computer algorithm, crowdsourcing is a reliable way of determining labels for facial expressions since workers are accustomed to analyse facial expressions in their everyday lives. Barry Borsboom [56] created a crowdsourcing game that consists of a similar version of the board game "Guess who?", where players have some animated faces and have to pick one. The goal of the game is to be the first to determine which face the opponent has selected, while resorting to asking "yes or no" questions to his opponent. Based on the questions it's possible to collect the attributes of each facial expression in the database. The facial expression database used in this game was ADFES [57], which is filled with actor's facial expressions, meaning that these facial expressions are not genuine, i.e the actors were forced to perform those facial expressions.

Chen, Hsu, and Liao [58] proposed a method to leverage photos in social groups like Flickr and removing noisy photos through crowdsourcing, in order to acquire effective training photos for facial attribute detection. The collected results were very similar to those of manual annotations.

There are many other datasets with facial expressions. Table 2.1 (adapted from [59]) shows a comparison between the most popular facial expressions datasets. As presented in the table, the majority of the datasets are collected in laboratory, where emotions are induced through different methods, with videos being the most used tool. Unlike these datasets, our dataset was captured in a realistic scenario, where players were aware that their facial expressions can control the game.

References	Elicitation method	Size	Emotion description	Labeling	Accessibility
Kanade, Cohn, and Tian [60]	Posed	210 adults, 3 races;	Category: 6 basic emotions, and	FACS	Y
Sebe et al. [61]	Natural: Subjects watched emotion-inducing videos	28 adults	Category: Neutral, happy, surprise	Self-report	N
Pantic et al. [62], [63]	Posed: static images, videos recorded simultaneously in frontal and profile view; Natural: Children interacted with a comedian. Adults watched emotion-inducing videos	Posed: 61 adults Natural: 11 children and 18 adults. Overall: 3 races Available: 1250 videos, 600 static images	Category: 6 basic emotions, single AU and multiple AUs activations	FACS, Observer's judgment	Y
O'Toole et al. [64]	Natural: Subjects watched emotion-inducing videos	229 adults	Category: 6 basic emotions, pizze, laughter, boredom, disbelief	Observer's judgment	Y
Yin et al. [65]	Posed: 3D range data by using 3DMD digitizer.	100 adults Mixed races	Category: 6 basic emotions. Four levels of intensity	Observers' judgment	Y
Gunes and Piccardi [66]	Posed: two cameras to record facial expressions and body gestures respectively	23 adults Mixed Races Available: 210 videos	Category: 6 basic emotions, neutral, uncertainty, anxiety, boredom	N/A	Y
Chen [67]	Posed	100 adults, 9900 visual and AV expressions	Category: 6 basic emotions, and 4 cognitive states (interest, puzzle, bore, frustration)	N/A	N
Roisman, Tsai, and Chiang [68]	Natural: subjects were interviewed to describe the childhood experience	60 adults Each interview last 30-60min	Category: 6 basic emotions, embarrassment, contempt, shame, general positive and negative	FACS	N
Bartlett et al. [69]	Natural: subjects were tried to convince the interviewers the were telling the truth	100 adults	Category: 33 AUs	FACS	N
SAL [70]	Induced: subjects interacted with artificial listener with different personalities	24 adults 10h	Dimensional labeling/categorical labeling	FEEL-TRACE	Y
Douglas-Cowie et al. [71]	Natural clips taken from television and realistic interviews with research team	125 subjects, 209 sequences from TV, 30 from interview	Dimensional labeling/categorical labeling	FEEL-TRACE	Y

Table 2.1: A comparison between the most popular facial expressions datasets [59].

2.5 Summary

In this chapter we presented an overview of the most relevant crowdsourcing research fields. The works described show us that crowdsourcing is a good source for collecting reliable and large amounts of data, although we believe that crowdsourcing is only on its early stages and a lot of work is needed in some of its research fields. One of these fields is the crowdsourcing models, where only few models are capable to handle with multi-level judgements. In addition, we believe that for problems where the aim is to infer the true label of an image, visual features can be useful. However, only Raykar uses it.



Crowdsourcing Facial Expressions Labels

3.1 Building a Real-World Facial Expressions Dataset

We resorted to CrowdFlower site to collect labels for a facial expressions dataset – the NovaEmotions dataset [1]. Figure 3.1 presents some examples of these facial expressions which were collected through a gamification process where players affectively interact with a game. The objective of this game is to perform the facial expression challenged by the game. While players try to perform the facial expression a score of the current facial expression is represented to the player. The player who better performs this *challenged* facial expression wins the round. In total, more than 40,000 facial expressions were collected through the NovaEmotions game.

Figure 3.2 depicts the entire process used to collect the dataset. In the left side, is represented the gamification process to collect the facial expressions and described in [1]. This chapter describes how these real-world facial expressions were annotated via a crowdsourcing process. The design of the crowdsourcing process included the worker’s interface and the attributes that directly influenced the results quality. The correct values of the crowdsourcing process attributes were also estimated through experimentation. After collecting the labels, we analysed the worker’s agreement to understand how reliable are these crowdsourcing labels.

In this chapter, we will present the process to collect crowdsourcing labels for this dataset, the NovaEmotions. The design of the crowdsourcing job was carefully planned:



Figure 3.1: Example faces from the dataset.

we obtained several judgements per image, each facial expression was linked to an intensity, and different worker selection criteria and batch jobs were inplace to reduce bias. This effort is also particularly relevant because, the annotations of a facial expression image will not be a binary label, but a distribution across the different expression labels. This is extremely useful for creating better affective-interaction models.

3.2 Optimizing Intra-Worker Agreement

The main objective of a crowdsourcing task is to obtain a set of label that all annotators agree with. An estimated label that is consensual among all annotators has more probability of being correct than a non-consensual one. The agreement for the image i is the number of votes for the winner label over the total of votes for the image i , as presented in 3.1. Thus, we will choose the label l_{ij} that maximizes the agreement for each image i

$$Agreement(i) = \frac{1}{|\mathcal{L}_i|} \sum_{j \in \mathcal{L}_i} \mathbb{1}_{y_i}(l_{ij}). \quad (3.1)$$

This corresponds to a majority voting approach, where the estimated label of an image i is the most voted label. In this chapter, we will use the majority voting technique to estimate the true labels of an image and in chapter 4 other models that try to improve the overall judgements quality will be examined.

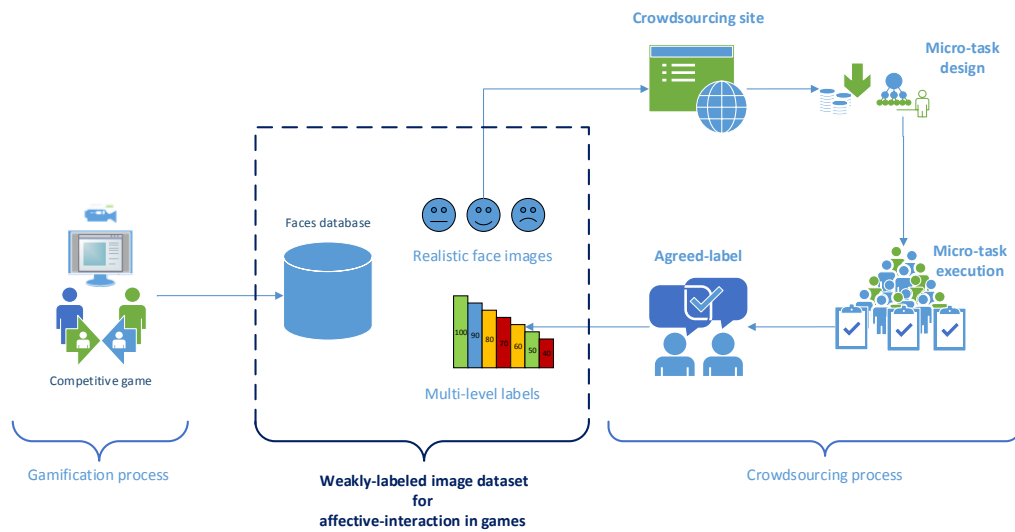


Figure 3.2: The gamification and crowdsourcing processes to generate the facial expressions dataset.

3.3 Crowdsourcing Task Design

In this section we identify the key factors that affect crowdsourcing results, and also present the experiments that led to the most reliable values. These factors can be divided into two groups, worker qualification and job attributes.

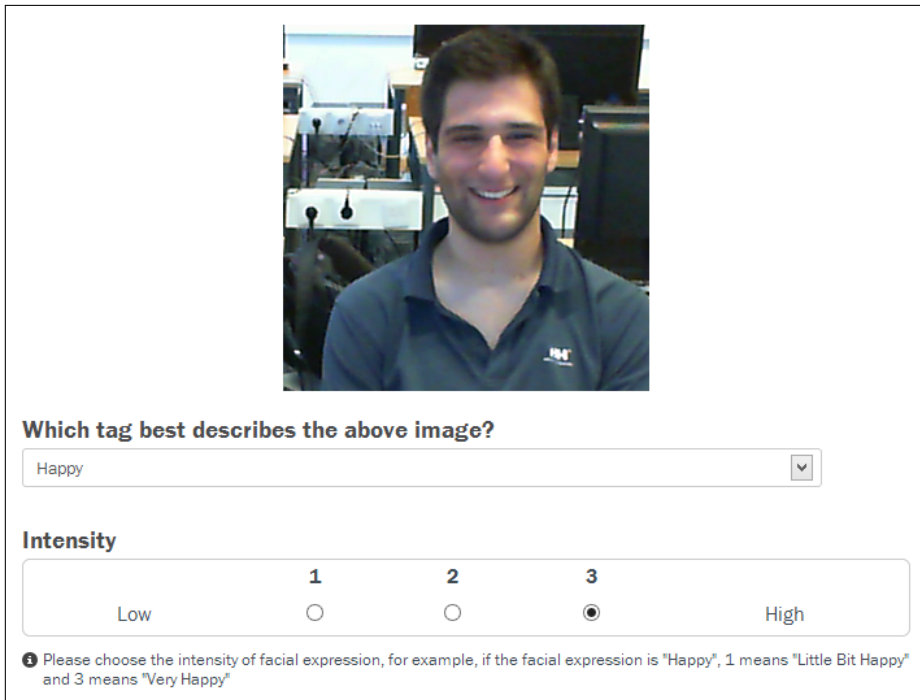
The worker interface is presented in Figure 3.3. In this interface the worker must (1) explicitly select the player's facial **expression**, and (2) select the **intensity** of the facial expression. This second action is intended to disambiguate and quantify the certainty of a worker's annotation. It is important to limit the set of choices and when possible not include text fields because this leads to open responses problems when matching labels. When building the interface we followed closely two important design principles:

1. The interface must be as simple as possible, a cluttered interface will lead to workers' fatigue.
2. The workers' interface should not require much interaction. This not only helps preventing workers' fatigue, but also allows to save time and consequently money.

As we shall see, the crowdsourcing task design goes much beyond the visual aspect of the workers interface.

3.3.1 Answers domain

The answers domain needs to be as short as possible for two reasons: (1) A high number of answers will demand more attention from the worker which contributes to her boredom. (2) The posterior data analysis will be easier. The answers must be clear and can not be ambiguous. For example, one can not have the answer *Happy* and the answer *Smiling*. Our answers domain will be the following facial expressions labels: *Angry*, *Contempt*,



Which tag best describes the above image?

Happy

Intensity

Low 1 2 3 High

ⓘ Please choose the intensity of facial expression, for example, if the facial expression is "Happy", 1 means "Little Bit Happy" and 3 means "Very Happy"

Figure 3.3: Worker interface.

Disgust, Fear, Happy, Neutral, Sad, Surprise. To this list we added *Ambiguous expression* for images in which the facial expression is not entirely clear and also *Not a face*.

3.3.2 Selecting a Pool of Workers

To be accepted in a job, a worker must pass some qualification criteria. Due to the difference in skill between workers, we need to find the group of workers that best suits a given micro-task. In our crowdsourcing job we would like to rely on common knowledge to interpret facial expressions. Therefore, our worker qualification process is based on the following attributes:

Cultural background As shown in [30], facial expressions are not culturally universal. Each country has its own culture, customs and non-verbal communication traits. Therefore we needed to use a group of workers that were culturally close to the players of the game. To ensure this, it is possible to include or exclude countries from a list of allowed countries.

Limiting the number of judgements per worker The maximum number of judgements a worker is allowed to complete can be limited. A small number can significantly increase the duration of a job because more workers will be required, while a large number will require less workers, but it may be too tiresome and/or distracting for each worker after several micro-tasks.

Job	Micro-task/Page	Judgements	Workers	Cost/Micro-task	Agreement
#1	20	10	102	0.004 \$	70.69 %
#2	115	5	32	0.004 \$	70.79 %
#3	20	5	93	0.009 \$	72.81 %
Total	-	20	227	0.017 \$	69.18 %

Table 3.1: Job’s parameters.

3.3.3 Job Attributes

Besides the worker’s cultural background and maximum amount of allowed judgements, there are other finer-grain job attributes that must be parametrized. The attributes that mostly influence the job’s quality and cost are:

Gold questions The quality of a worker can actually be determined during a micro-task. For that, one must provide gold questions where the worker is tested with pre-labelled images. The worker must answer correctly to at least four gold questions, before performing further micro-tasks.

Price per micro-task The price per micro-task is the most difficult attribute to estimate. This is correlated with other factors such as: worker’s country or task difficulty. Since the price per each micro-task can make the job cost increase significantly, one must define the price-quality ratio wanted.

Minimum number of judgments A job is divided into several pages and each page has a predefined number of micro-tasks. Usually a worker completes at least one page, making this parameter also work as the minimum number of judgements a worker must complete.

Judgements per image In this case, a micro-task is an image that must receive a given number of judgements. The larger the number of judgements, the greater the confidence of our task design. At least five judgements per image were collected in each job.

3.4 Tuning Jobs

In this section we present the results collected through crowdsourcing with a subset of our dataset. By running these experiments, we were able to estimate the best parameters that should be used for the full dataset. We ran three jobs, in a non-parallel way, to find the best parameters. Each job contributed to understand how one parameter should be set. The submitted jobs can be seen in Table 3.1. In total we run 3 tuning jobs. In the way CrowdFlower works, workers involved in a job were not the same as in other jobs. For that reason, one job may have better workers than the other job and consequently achieve better results.

3.4.1 Setup

We collected a sample of 500 images from our dataset and used this subset in all tuning jobs. The considered images had an equal distribution of labels, and an automatic algorithm was capable of detecting them correctly.

As explained above, cultural differences were considered by selecting only English speakers in Europe, US and Australia. The first parameter to tune is the number the judgements because this works as a multiplier in the final price to pay. Therefore, finding this parameter in the first job allows us not to spend more than necessary in the following jobs. Moreover, this value can be easily tune by requesting a large number of judgements and posteriorly analyse how many judgements were needed to converge to a label. This trick can also be used to find the best maximum of judgements per worker: we did not limit the number of micro-tasks per worker and posteriorly we analysed how many judgements an average worker can perform without decrease his performance. For the other parameters, we used values of reference based on literature. Thus, we choose to have 20 micro-tasks per page which allows workers to quit every 20 micro-tasks performed and pay 0.004\$ for each completed micro-task.

3.4.2 Number of Judgements per Image

In the first job, we wish to find how many judgements an image needs to achieve a consensual result. By collecting 10 judgements per image we guarantee that we have a larger than the optimal number of judgments.

Figure 3.4 shows the evolution of annotations agreement as votes arrive for each facial expression. In these charts we plot the average agreement for all images of each facial expression. The green line shows the best sequence of votes to keep a high agreement (the first incoming votes are consensual). The red line shows the worst sequence of votes, in other words, the sequence of votes that keep the agreement at a low value. The blue line is a random sequence of votes and is the line that is most likely in a realistic scenario. Let y_i be the estimated label for the image i and q the facial expression that we want to analyse. Considering the set $S_q = \{i \in \mathcal{I} : y_i = q\}$. We can model the annotating process as a sequence of votes as follows:

$$f_q(x) = \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} \frac{1}{x} \sum_{j=1}^x \mathbb{1}_{y_i}(l_{ij}) \quad (3.2)$$

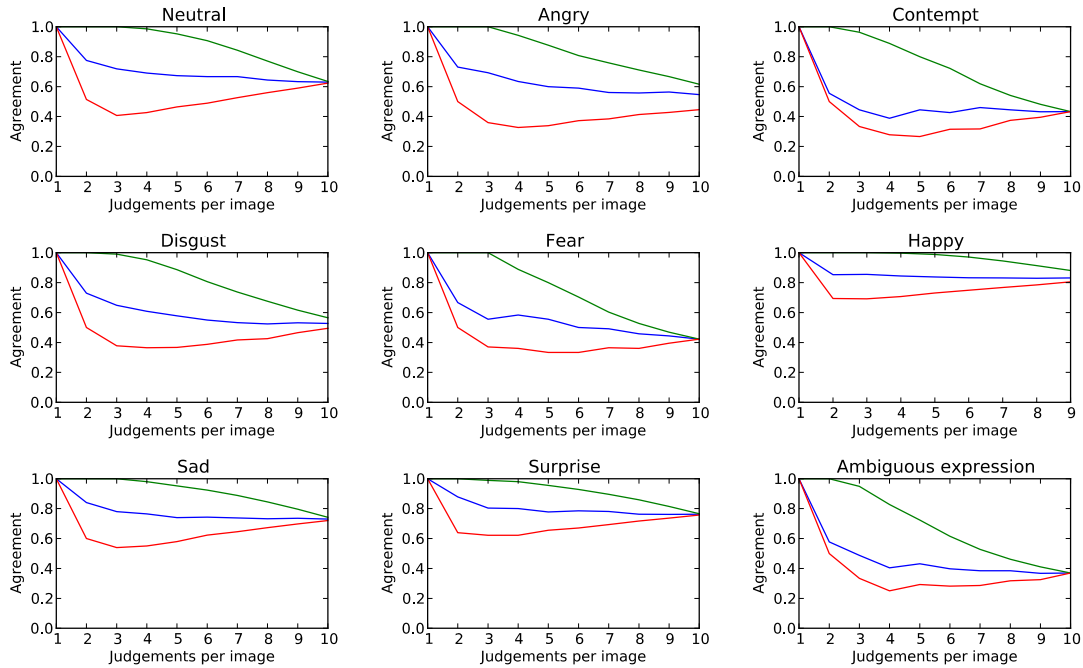


Figure 3.4: Average of image’s agreement over number of judgements. The blue line is a random sequence of judgements per image, the red line is the worst sequence, and lastly, the green is the best sequence.

The mainly objective of this tuning job is to infer the optimal value of judgements per image. The results shows us that the agreement converges almost for every facial expression when the number of judgements is 5. Therefore, we do not need to pay for 10 judgements and consequently duplicate the costs. The Figure 3.4 also shows us the difference of judgements between all facial expression. The area between the green line and the red line, shows the discrepancy of judgements between workers, so the area increases with the number of different labels voted. For example, the facial expression *happy* has the smallest area, which means, *happy* is confused with few labels (*neutral* mostly). It is interesting to note that the facial expressions *happy* and *surprise* only need, approximately, 3 votes to stabilize at an agreement’s value. If we could set a constraint at CrowdFlower to stop collecting judgements from an image that already has 3 judgements and is *happy* or *surprise* we could decrease the job cost without decreasing the results quality.

In summary, Figure 3.4 shows that we do not need a large number of judgements per image to infer the best estimated agreement. In the majority of facial expressions, the agreement stabilizes with only 5 judgements.

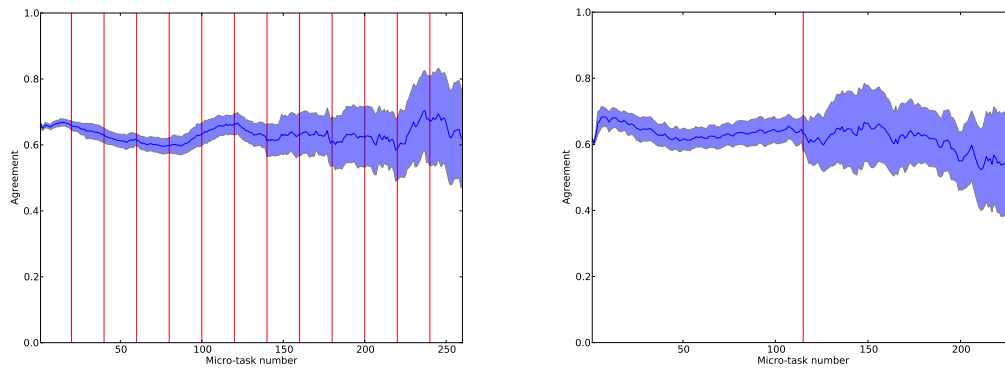


Figure 3.5: The blue line is the average of agreement for the n th judgement of each worker. Each red dot is the agreement of one worker for the n th judgement. The area around the blue line is standard deviation. On the left side, is presented the analysis of the first job where workers. On the right side.

3.4.3 Number of Judgements per Worker

For this experiment we changed the number of micro-tasks per page. As we can see in Table 3.1 we increased this value from 20 to 115. Each worker will complete several sets of 115 micro-tasks per set, being 115 the minimum number of judgments a worker must complete. To assess workers endurance over a large number of micro-tasks per page, we plot the mean of workers' agreement over time on Figure 3.5. On the left side is presented the first job where workers had to perform 20 micro-tasks per page, whereas on the right side is presented the second job where workers had to perform 115 micro-tasks per page. As we can observe, for the highest value of micro-task per page (right picture), the line on the chart decays for workers who perform more micro-tasks. In contrast, small values of micro-tasks per page (left picture), show us that workers with more micro-tasks are those who perform more reliable results.

As expected, there is a significant number of workers who perform fewer micro-tasks. Therefore, the standard deviation increased over time in both jobs. Comparing the standard deviation in both jobs, the second job had less workers than the first one, so, the standard deviation has increased in global. This decay in agreement in the second job is not a coincidence, because we can notice that this decay starts around the 116th micro-tasks which is the start of a new set of micro-tasks. So, we can conclude that a minimum judgements per image achieve better results and workers whose performs more judgements are better. Therefore, we will not limit the worker's judgements and we will use the value of 20 micro-task, mainly because this allow the workers to quit the job as soon as they want because each page has few micro-tasks and with this ensure the focus and commitment of each worker. This will avoid situations like the second job where workers starts a new set of micro-tasks but got bored in the middle of it.

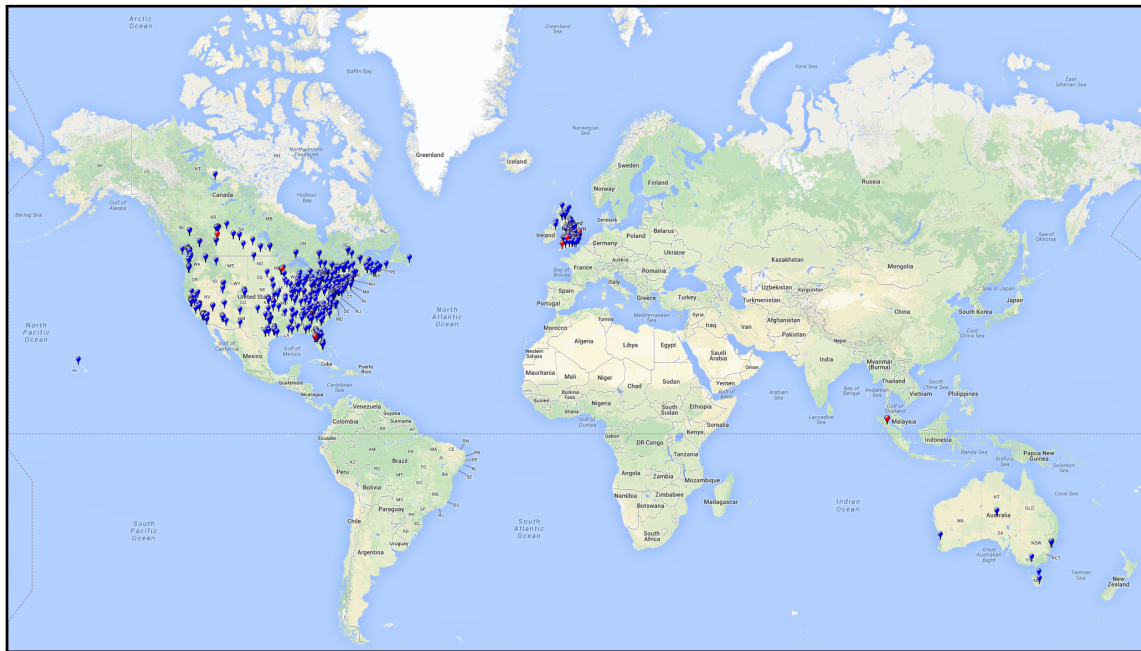


Figure 3.6: Worker's location

3.4.4 Worker's Payment vs Geographic Location

The worker's payment is an important parameter since it is related to the worker's motivation. It is important to highlight that the amount paid to each worker is strongly related to the complexity of the micro-task and the time required to complete it. Another factor that is important to consider is the worker's location. We must be aware that users from different countries have reward expectations. Therefore, if we hire workers from developed countries we have to pay more than to a workers from developing countries. In our case, we only hire workers from countries which have English as the native language and were in Europe, Australia or in the US.

In our case, our task presented a very low difficulty - the worker only had to identify a person's facial expression in an image. This is almost an instantaneous task and requires only a few seconds to complete. Therefore, we set up the price to 0.004 \$ in the first two jobs. The third tuning job had the objective to infer the optimal value to pay for each completed micro-task. We increased the price for more than 100 % and as expected, we achieved better results than the initial two jobs as show in Table 3.1, approximately 2%. We believe it is not worthy to have the double of the costs for an improvement of 2% in accuracy. Thus, to secure a final agreement above 70% we increased the initial values by 50%, paying each worker 0.006 dollars per judgement.

3.5 Full Jobs: Results and Discussion

In this section we present the results of collecting the facial expressions of each image of our dataset. The dataset has over 40,000 images and was randomly divided into four separate jobs with approximately the same number of images. The parameters used were: 20 micro-tasks per page, 5 judgements per image and 0.006 dollars per micro-task and no limit on the number of judgements per worker. In total, we paid to approximately 1100 workers and the location's workers can be seen in Figure 3.6. The following analysis and discussion considers the four jobs as an whole.

Golden images The images used in the gold questions had the best agreement in the jobs described in Section 3.4. We selected the images with an agreement above 90%, which makes a total of 115 golden images. Although the majority of crowdsourcing sites has some control policy to identify bad workers, gold questions are very important to assert about a worker skills and commitment to the task.

Collected votes The Table 3.2 presents the distribution of votes collected for all dataset. On the first column is presented the answer domain. In other words, each worker had to choose one of this answers when performing a micro-task. In each middle column is presented the number of votes collected in each job (*Votes*) and the how many times the answer achieved a majority of votes (*Win*). In the last column is presented the distribution of labels inferred by majority vote.

Due the player's commitment when playing the game, sometimes the facial expression performed was very different than the challenged one. For example, when one player was challenged to perform the facial expression *sad*, the people surrounding him laughed and consequently she lose focus and laughed as well. So, as expected ,the majority of facial expressions is *happy*. On the other hand, the label with less votes was *not a face* with only 46 labels on our dataset. In Figure 3.7 is an example of two images with a *not a face* label.



Figure 3.7: Example of two images whose the label is *Not a face*.

	Job 1		Job 2		Job 3		Job 4		Total		Distributions
	Votes	Win	Votes	Win	Votes	Win	Votes	Win	Votes	Win	
Not a face	244	7	193	9	231	14	187	16	855	46	0.11 %
Angry	995	130	978	131	852	105	822	94	3647	460	1.12 %
Sad	4251	704	4004	687	4013	687	4220	703	16488	2745	6.70 %
Neutral	8608	1955	8623	1958	10091	2298	9404	2157	36726	8359	20.40 %
Disgust	2885	535	3448	704	2909	548	2989	593	12231	2380	5.81 %
Surprise	10138	1860	10187	1862	9934	1865	10462	1879	40721	7382	18.01 %
Fear	1116	116	1009	109	928	93	1011	116	4064	434	1.06 %
Ambiguous	2316	258	2245	252	1764	188	2301	261	8626	959	2.34 %
Contempt	1736	134	1735	130	1448	112	1747	134	6666	510	1.24 %
Happy	25368	4600	24862	4489	24922	4430	24408	4404	99560	17707	43.21 %
Total	57657	10299	57284	10331	57092	10340	57551	10357	229584	40982	100.00 %

Table 3.2: Number of votes of each facial expression for each job.

	Neutral	Angry	Contempt	Disgust	Fear	Happy	Sad	Surprise	Ambig.	NAF
Neutral	0.67	0.11	0.19	0.05	0.05	0.04	0.08	0.02	0.10	0.14
Angry	0.02	0.50	0.04	0.04	0.02	0.00	0.02	0.01	0.02	0.01
Contempt	0.06	0.08	0.42	0.05	0.03	0.01	0.03	0.01	0.06	0.03
Disgust	0.03	0.11	0.08	0.60	0.11	0.01	0.05	0.02	0.09	0.04
Fear	0.01	0.02	0.02	0.03	0.52	0.00	0.02	0.03	0.02	0.02
Happy	0.09	0.04	0.07	0.06	0.04	0.88	0.02	0.05	0.09	0.10
Sad	0.05	0.07	0.08	0.07	0.05	0.01	0.73	0.01	0.04	0.02
Surprise	0.02	0.02	0.03	0.04	0.15	0.02	0.01	0.83	0.08	0.06
Ambiguous	0.04	0.05	0.06	0.06	0.04	0.02	0.04	0.03	0.47	0.12
Not a face	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.45

Table 3.3: Confusion matrix for each facial expression

3.5.0.1 Facial-expression

The confusion matrix determined by majority vote is presented in Table 3.3. This matrix illustrates the judgements confusion among the nine alternatives: all six basic expressions, the composed expression *contempt*, an ambiguous and a noisy capture. Each row of the matrix represents the worker’s votes for each facial expression, whereas each column represents the actual label achieved by majority vote.

The diagonal of the confusion matrix illustrates how the majority of expressions are clearly separable from the others. The facial expressions *happy* and *surprise* were the most consensual among all workers with an agreement of over 0.8. Many facial expressions are confused with *neutral* due the intensity of the facial expression. One worker may consider a person grinning as *happy* while another worker may consider just *neutral*. The most dubious facial expression is *contempt* which is often confused with *neutral*, once more due the intensity of expression.

Ambiguous expressions achieved a surprising agreement of 0.47 because it was confused with *neutral* 10 % of the time. This means that when an user is not performing one of the other facial expression, some workers assign the *neutral* label while others assign the *ambiguous* label and the remaining workers try to choose a label. So, we can conclude that workers follow different decision criterion when they are faced with ambiguous expressions.

Expression				
Neutral	██████████	██████████	██████████	██████████
Angry	██████████	██████████	██████████	██████████
Disgust	██████████	██████████	██████████	██████████
Fear	██████████	██████████	██████████	██████████
Happy	██████████	██████████	██████████	██████████
Sad	██████████	██████████	██████████	██████████
Surprise	██████████	██████████	██████████	██████████
Contempt	██████████	██████████	██████████	██████████
Ambiguous	██████████	██████████	██████████	██████████
Not a face	██████████	██████████	██████████	██████████

Table 3.4: Workers’ votes for facial expressions with lowest and highest agreement.

In Table 3.4 we present some examples of facial expressions with low agreement in our dataset. The first image is confused with many labels (all except *angry* and *not a face*). It is odd to observe that, although, many workers had voted in different tags, few voted on label *ambiguous expression*. A closer inspection of Table 3.4 gives a good insight to how

workers annotated images: they actually tried to make good decisions, since it is evident that there are a bias on each image (some labels received no votes, or votes went to a two or three different labels).

Agreement. An image with agreement of 1.0, means that all of 5 votes were on same label, this happens on approximately 40 % of our dataset and 62.5 % has at least 0.8 of agreement, which means 4 of 5 votes were on the same label. On the other hand, the dataset has 1.1 % of images with agreement of 0.2, in other words, all the votes were in different labels. Although there are images with low agreement, this exploits the advantage of *not* using the binary judgement model. For example, we can conclude that facial expression with low agreement is an *ambiguous expression*. Another way to take advantage of images with low agreement, is use them as a counter-example. For example on an image with 0.2 of agreement, we can not infer the correct label. Although, if that image doesn't have any vote as being *sad*, one can conclude that definitely the player's facial expression is not *sad*.

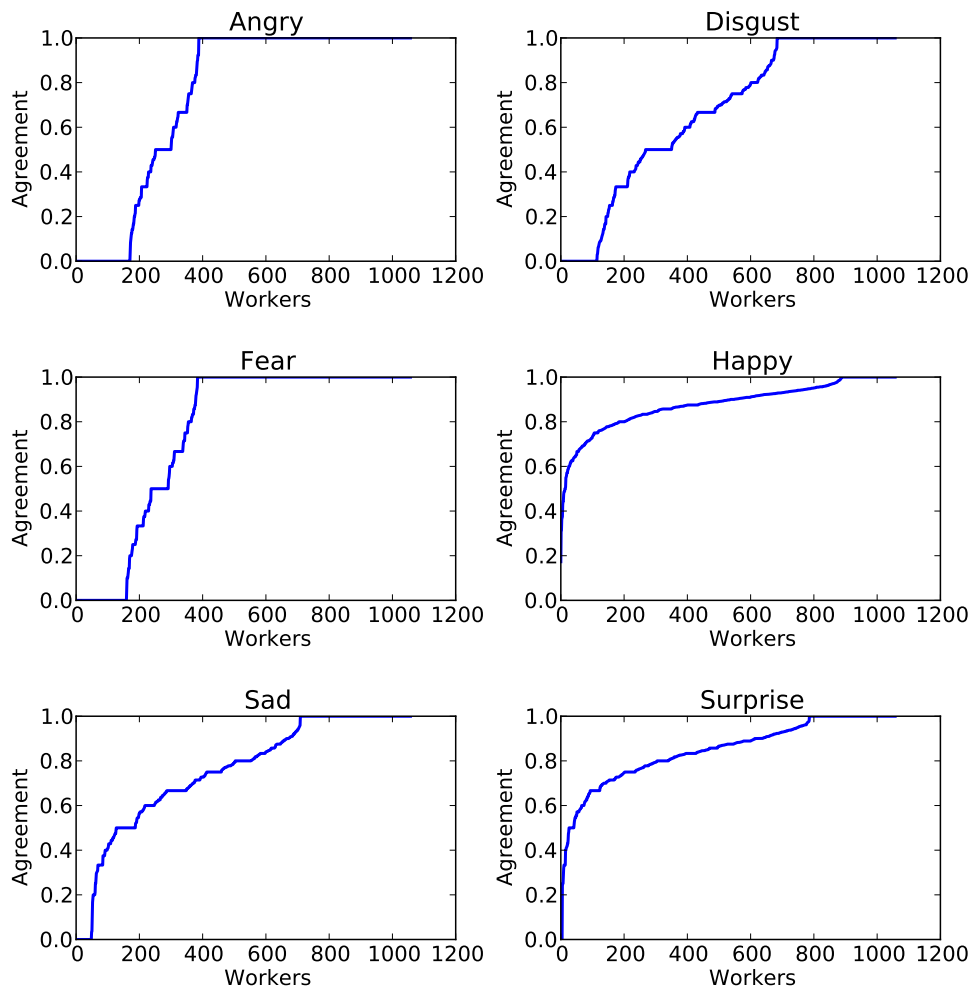


Figure 3.8: Workers agreement with the selected label, sorted by agreement

3.5.0.2 Per-expression Analysis

Figure 3.8 illustrates the distribution of workers' agreement over each facial expression sorted by agreement. We define the set of all labels of a worker for a given facial expression q as follows

$$\Omega_q = \{l_{ij} \in L^j : y_i = q\} \quad (3.3)$$

where the estimated facial expression y_i is equal to q where q is a given facial expression. Then we compute the agreement between this and the labels that each worker labelled correctly (according to the majority) as follows:

$$f_q(j) = \frac{|\{l_{ij} \in \Omega_q : l_{ij} = y_i\}|}{|\Omega_q|}. \quad (3.4)$$

$$\phi_q = \{f_q(j) : j \in J\} \quad (3.5)$$

Then we sort the results produced by Equation 3.5 and plot them. We apply this function for all workers and sort the results. It is interesting to observe the shape of these curves - ideally they should all start and end with an agreement of 1.0 (meaning that all workers agreed on one single label for every image). The area underneath the curve indicates the overall labelling agreement across all workers for that expression.

The line smoothness is determined by the number of workers with agreement between 0.0 and 1.0 (exclusively). There are two factors that contributes to the line smoothness: (1) The number of instances to label of that facial expression, if one worker only labelled one instance of some facial expression the agreement will be 0.0 or 1.0. (2) The workers expertise, ones that only produce good results (agreement 1.0) and ones that only produce bad results (agreement 0.0).

The line starting point is explained by the lack of expertise of some workers or the difficulty to label some facial expressions. The agreement curves for *sad*, *angry*, *fear*, *disgust* and *surprise*, show that some workers had an agreement of 0.0, which means that these workers failed all images for this expression. The exception to this trend occurs for the expression *happy* where the worst worker agreement was near 0.2.

3.6 Summary

This chapter describes the crowdsourcing job design for annotating real-world facial expression images with the correct facial expression. In this chapter we presented the crowdsourcing job attributes that were considered when designing a crowdsourcing job. We run 4 jobs with a small sample of our dataset to tune each one of these attributes: (1) judgements per image; (2) The maximum judgements per worker; (3) Price per micro-task. This process was the objective to maximize the agreement across workers' judgements.

In a first job we collected 10 judgements per image. This allowed us to infer that we only need 5 judgements to achieve the same agreement. In a second job, we study the impact of using a large number of micro-tasks per page. This approach forces the worker perform various sets of micro-tasks. Therefore, the workers commitment decays over time due the, probable, worker's boredom. Last but not least, we duplicated the cost of the whole job and only increased the data quality by 2%. This allowed us to conclude that the price per micro-task also increased the data quality but we believe does not worth having the double of costs for such small increase.

After the tune jobs, we proceed to label all our dataset. This results in a dataset with over 40,000 images player's facial expression and five judgements per facial expression [5]. It is important to note that all judgements for the full set of images are provided to foster the investigation of other relevance models for affective-interaction and for crowd-sourcing models, like the ones that we study in the next Chapter. This dataset can be downloaded from <http://novasearch.org/datasets/>.

4

Benchmarking Weak-labels Combination Strategies

4.1 Crowdsourcing Methods

Weakly-labeled learning is a problem where the image labels of the target domain are not entirely reliable. There are many factors that can contribute to this: an annotator may not be familiar with the context of the dataset or the image is very hard to label. Also, annotators have their own bias. For these reasons, there are techniques that try to infer the true label of an image using weak-labels. In this chapter we will evaluate six popular techniques to achieve this kind of task. In the following sections we detail each crowdsourcing method and then Section 4.2 presents the evaluation of the studied crowdsourcing methods. In Section 4.3 we perform a controlled experiment to better understand how the crowdsourcing methods are affected by the expertise of workers. Section 4.4 presents experiments with real workers. In the last Section (4.5) we will add two types of noisy workers to a real crowdsourcing dataset: random and adversarial workers.

4.1.1 Majority (MV) and ZenCrowd (ZC)

The majority voting is the simplest process to merge the labels collected from different workers. In this method, the estimated label is the one with more number of votes. ZenCrowd [44] extends this method by modelling the workers' expertise. The authors estimate the workers expertise and the true labels jointly by running the EM algorithm.

4.1.2 Dawid and Skene (DS)

In 1978, Dawid and Skene were the first to propose a model where we can estimate a *true* label when we have weak annotations. In their article [40], Dawid and Skene try to solve the problem where a patient is observed by different clinicians and thus, different diagnosis are produced for the same patient. There are reasons that can lead to a non-consensual diagnosis: The same question asked to the same patient, but by different clinicians can have different responses; different clinicians can have different interpretations of the same response; different observers can have different background knowledge.

The authors characterize the probability of an observer j vote c when the *true* class is q as π_{cq}^k . These probabilities are called the individual error-rates of each observer. Ideally, this probability is always 1 when $c = q$ and 0 otherwise, which means that the observer j always makes a correct diagnosis. However, this optimal scenario doesn't exist in real life. The first objective of the authors is to estimate these error-rates. For this, they consider n_{ic}^j the number of responses c that observer j receives from the patient i . This is important to consider because an observer with more feedback from his patient is more likely to perform a better diagnosis than an observer with less feedback. They also consider T_{ic} as indicator variables. Ideally, $T_{ic} = 1$ when c is the true label and 0 otherwise. The algorithm from Dawid and Skene runs as follows:

- Take initial estimates of T_{ic} for example using the majority voting system.
- Estimate π_{cq}^k and p_j using the equations in 4.1 and 4.2, respectively.

$$\pi_{cy}^k = \frac{\sum_i T_{ic} n_{il}^k}{\sum_l \sum_i T_{ic} n_{il}^k} \quad (4.1)$$

$$p_c = \sum_i T_{ic} / I \quad (4.2)$$

- Estimate T_{ic} using the data previously estimated using Equation 4.3

$$p(T_{ij} = 1) = \frac{\prod_{k=1}^K \prod_{l=1}^J (\pi_{jl}^k)^{n_{il}^k} p_j}{\sum_{q=1}^J \prod_{k=1}^K \prod_{l=1}^J (\pi_{ql}^k)^{n_{il}^k} p_q} \quad (4.3)$$

- Repeat step (2) and (3) until convergence.

This problem is very similar to our problem, where we can see the observers as our workers. Although crowdsourcing workers do not have any patient to observe, they have to observe an image and make a judgement which has the same concept as the observer/patient relation.

4.1.3 GLAD

Later, Whitehill et al. [41] proposed the GLAD model (Generative model of Labels, Abilities and Difficulties). This model assumes that each image j has its own difficulty β to label and also that each annotator i has an expertise level α . In Figure 4.1 is represented the graphical model of GLAD where the label of the annotator j for the image i , l_{ij} , is dependent of the annotator's expertise and image difficulty. Then the authors assume the following model to estimate the true label of an image (z_j):

$$p(l_{ij} = z_j | \alpha_i, \beta_j) = \frac{1}{1 + e^{-\alpha_i \beta_j}} \quad (4.4)$$

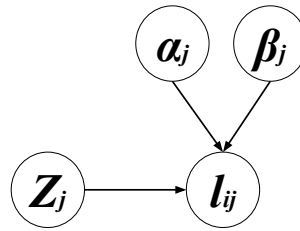


Figure 4.1: The graphical representation of GLAD model.

Under this model the authors followed an EM approach to estimate these parameters as well as the true label. GLAD showed good results and was adopted as a baseline in later models or extended to support other attributes. One of such models, is the CUBAM model which we will explain in the next section.

4.1.4 CUBAM

An extended version of GLAD model was proposed by Welinder et al. [42]. This model, seen in Figure 4.2, introduces a high-dimensional concept of image difficulty and annotator expertise along with others attributes.

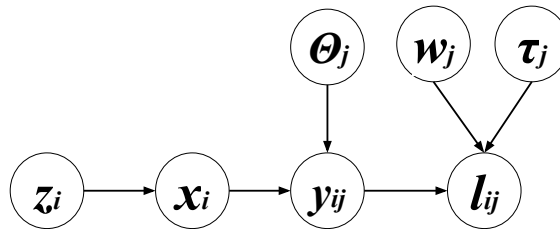


Figure 4.2: A simplified version of the graphical representation of CUBAM model.

The instance is virtually represented as a vector of task-specific measurements x_i which can be interpreted as an image's representation on the visual system of a ideal annotator. However, each annotator has her own interpretation, therefore access to a modified version of x_i , that can be represented as $y_{ij} = x_i + n_{ij}$ where n_{ij} is the noise produced by the individuality of each annotator. This vector y_{ij} is compared to vector

w_j which represents the expertise of annotator j in each component. In other words, this model finds the "areas of strengths" of each annotator unlike the majority of models in which the annotators are parametrized with a scalar value that simply indicates if an annotator is good or bad. The scalar projection of $\langle y_{ij}, w_j \rangle$ is compared to a threshold T_j . If this projection is above the threshold, the annotator assigns a label $l_{ij} = 1$ or $l_{ij} = 0$ if otherwise. That is represented by the following equation, where (ϕ) is a cumulative standardized normal distribution, a sigmoidal-shaped function:

$$p(l_{ij} = 1 | x_i, \sigma_j, T_j) = \phi \left(\frac{\langle w_j, x_i \rangle - T_j}{\sigma_j} \right) \quad (4.5)$$

4.1.5 Raykar (RY)

Raykar et al. [43] proposed a method of modelling the annotator expertise. They define the annotator expertise to label each binary label. *Sensitivity* α_j is the bias of an annotator labelling an image if the true label is one, whereas *specificity* β_j is the bias of labelling an image if the true label is zero. This can be defined as follows:

$$\alpha^j = p(l_i = 1 | z_i = 1) \quad (4.6)$$

$$\beta^j = p(l_i = 0 | z_i = 0) \quad (4.7)$$

Different annotators have different levels of expertise. In order to give more importance to workers who have more expertise, the authors considered beta priors for sensitivity and specificity:

$$p(\alpha_j | a_1^j, a_2^j) = \text{Beta}(\alpha_j | a_1^j, a_2^j) \quad (4.8)$$

$$p(\beta_j | b_1^j, b_2^j) = \text{Beta}(\beta_j | b_1^j, b_2^j) \quad (4.9)$$

In a similar way the authors assume a beta prior for the positive class $\text{Beta}(p | p_1, p_2)$, p is called the prevalence of the positive class.

4.2 Experimental Setup

As shown in the previous section, there are many crowdsourcing methods which propose to outperform the majority vote. However, designing an efficient crowdsourcing method is not enough to ensure the quality of the results. This quality is dependent on many factors, such as: (1) the job difficulty; (2) the incentives (monetary or not) and, most importantly, (3) the workers expertise. In this section we will study the behaviour of state-of-the-art crowdsourcing methods.

Algorithm 1 Benchmarking crowdsourcing models

Inputs:
workers_labels \leftarrow Set of worker labels
ground_truth_labels \leftarrow Set of ground-truth labels
random_workers \leftarrow Set of indexes of random workers.
for all *judgements* $\leftarrow 1$ **to** *max_judgements* **do**
 for all *model* \in *models* **do**
 for all *run* $\leftarrow 1$ **to** 30 **do**
 model_labels \leftarrow *model*(*workers_labels*, *judgements*, *random_workers*)
 acc \leftarrow *accuracy*(*model_labels*, *ground_truth_labels*)
 *run_accs*_{*model*,*judgements*} \leftarrow *run_accs*_{*model*,*judgements*} \cup {*acc*}
 end for
 end for
end for

The experiments described in this chapter will follow the procedure presented in Protocol 1. To measure the performance of these models, we will compute the accuracy of the estimated labels against ground-truth.

For each crowdsourcing model we run it using different judgments per image: this will allow us to infer the best number of judgments to use. Additionally, we consider a pool of *random workers*, which model the annotation process with a given type of random noise. In particular, we will use this pool of random workers to add random workers to our process, thereby ensuring that we know the type of noise that is being added. After selecting all *random workers*, we must choose workers from the remaining pool of workers. We consider three methods for choosing worker labels: (1) randomly select a worker from the pool of random workers; (2) use the same amount of workers as the number of judgments; (3) Ensure that each worker contributes with the same number of judgments. We will use the last method because it more closely mimics a generic scenario.

On each run, the model will randomly choose labels from the pool of *workers labels*. Since workers have different levels of expertise, the result of labelling the same instance can vary depending on the worker to which the instance was assigned to. In order to minimize the effects of this random assignment, each task is executed thirty times, being the median value the result used as output. The reason we use the median value instead of the average is because it provides a more realistic estimation of what to expect from a crowdsourcing model. For example, if a crowdsourcing model always produces 0.0 or 1.0 of accuracy, exclusively, the average is 0.5. However, this value is not representative, for it is impossible to happen. The median value represents the most probable value to occur.

4.3 Synthetic Experiment: Modelling Workers Expertise

The workers expertise is strongly connected with the results quality. For this reason, we have to use many judgements per micro-task and expect bad and good judgements among them. A good crowdsourcing method has to identify which workers produce bad judgements and which do not, consequently estimating each worker's bias/expertise. However, it is difficult to validate such thing with real data: one must have a well defined popularity of workers, containing workers with different levels of expertise. Since quantify the worker's expertise is a hard task, this kind of workers population is difficult to get. To address this problem, we created a framework which allows us not only to create synthetic micro-tasks with different levels of difficulty, but also to create synthetic workers with different levels of expertise. This framework will allow us to study how crowdsourcing methods behave with different types of workers. In the following experiments we will use the following notation:

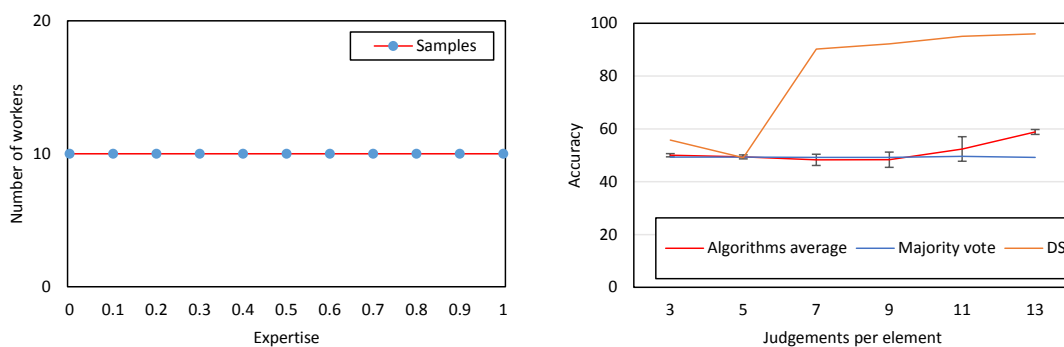
- **Synthetic element** - We will call synthetic element to each micro-task generated virtually. For the following experiments, we will use 1000 synthetic data.
- **Data difficulty** - A synthetic element has an associated level of difficulty. This difficulty is a random value between 0.0 and 1.0, where an element with difficulty 1.0 means that it is impossible to label correctly and 0.0 that is impossible to label incorrectly.
- **Categories** - We will perform binary experiments. This means that each synthetic element will have a binary label among two classes: 0 or 1.
- **Synthetic workers** - To classify our synthetic data we will create synthetic workers. This workers are not real workers. As synthetic elements, synthetic workers are virtually generated.
- **Worker expertise** - What distinguishes workers is the expertise level. This is a parametrized value between 0.0 and 1.0. Similarly to an element's difficulty, a worker with 1.0 of expertise can label everything correctly, whereas a worker with 0.0 cannot label any element correctly.
- **Judgements per image** - When requesting new judgements, the system will fetch the worker that has the least number of votes from the workers pool. The worker will label correctly if his level of expertise is bigger than the data difficulty.

4.3.0.1 Constant

We can now consider different populations of workers. Let us consider that we have a constant number of workers per level of expertise. The Figure 4.3(a) presents this type of population. The blue dots represent the samples used in the experiment. In total, we

created 11 groups of workers with different levels of expertise. Each group is formed by 10 workers. Thus, in total, this experiment has 110 workers to label 1000 instances with a diverse number of judgements per instance.

Although this not a realistic scenario, the objective of this experiment is to understand if the crowdsourcing models are able to identify the good and the bad workers. The results of running this experiment are presented in Figure 4.3(b). The average expertise of an worker is 0.5 and this was the obtained accuracy value. However, when the number of judgements is increased the accuracy also increases, but only by a slight margin. The difference between the majority vote and the remaining crowdsourcing methods are not relevant. However, DS outperforms all models with considerable margin (30%).



(a) Workers distribution

(b) Average of all models accuracy

Figure 4.3: Accuracy of crowdsourcing methods using a constant distribution of worker's expertise

4.3.1 Gaussian

As described in the previous experiment, using the same amount of workers for different types of expertises revealed similar results to choosing a label randomly. For this reason, we decided to make an experiment using a workers population which follows a normal distribution, presented in Figure 4.4(a). In this experiment, we will have many workers with expertises close to 0.5 and few bad and good annotators. Note that an expertise of 0.5 means that the worker have equally probability to label correctly and incorrectly. This is the same to randomly vote in a category. Nevertheless, in our setup, a worker's vote is connected to the instance difficulty.

The results presented in Figure 4.4(b) show us that a workers population in which the majority of the workers have fifty percent chance of classifying the instance correctly, the crowdsourcing workers are useless: the average is 50%, independently of the number of judgements. In addition, note that the standard deviation is almost non-existent. This means that, for all crowdsourcing methods, the result was practically the same. Comparing with majority vote, the difference is not so significant, but overall, majority vote achieves better results.

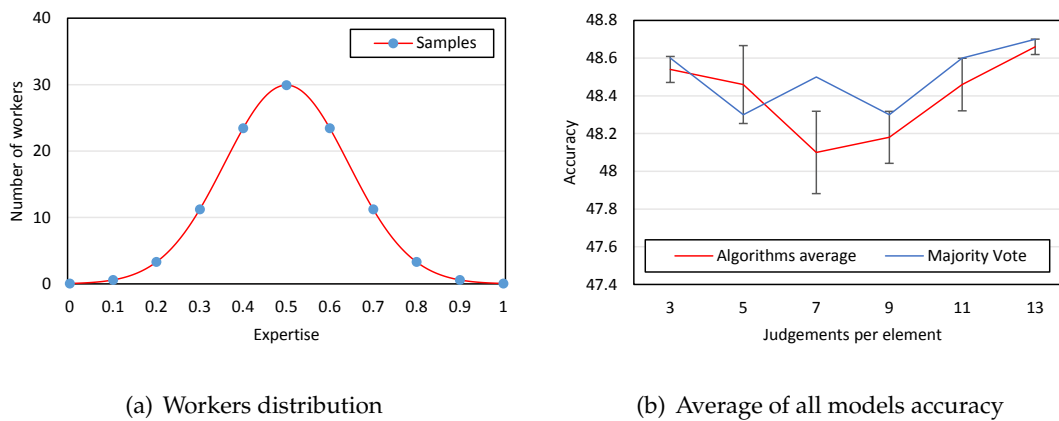
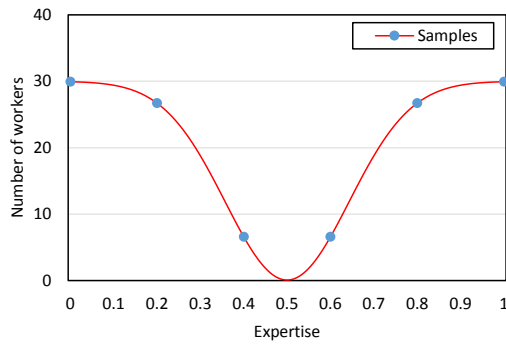


Figure 4.4: Accuracy of crowdsourcing methods using a normal distribution of worker's expertise

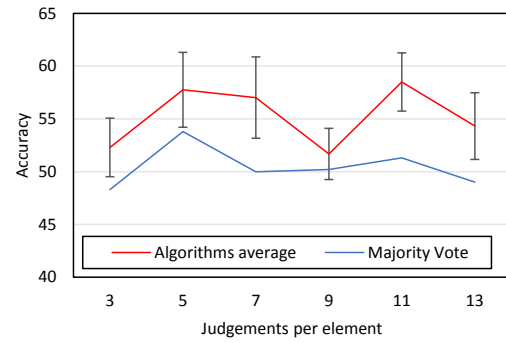
4.3.2 Inverse Gaussian

We can also consider the inverse of the above scenario. Instead of having a large number of workers with 0.5 of expertise, we consider a large number of bad workers and good workers, as presented in Figure 4.5(a). Once again, workers with an expertise of 1.0 always classify the data correctly, while workers with an expertise of 0.0 always classify incorrectly. Note that this kind of workers population is unlikely to happen. However, this experiment will allow to understand if crowdsourcing models can identify the few good annotators.

The results are presented in Figure 4.5(b). Unlike previous experiments, the average accuracy was not the same as the average workers' expertise. The average workers' expertise: while the average expertise in this population is 0.5, the average accuracy was always above that value. The best results were obtained for 5 and 11 judgements, where the accuracy was almost 60%. As the standard deviation suggests, some crowdsourcing methods performed better than other. In the Section 4.3.5 we will make an analysis of each method individually. However, we can observe that crowdsourcing methods perform better than the majority vote for a considerable margin (around 5%).



(a) Workers distribution



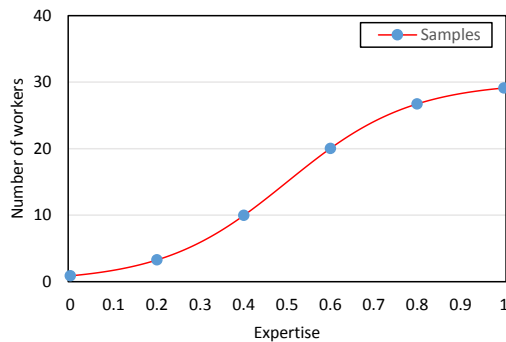
(b) Average of all models accuracy

Figure 4.5: Accuracy of crowdsourcing methods using a inverse Gaussian distribution of worker's expertise

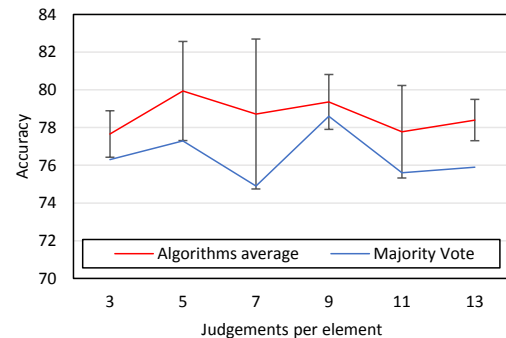
4.3.3 Logistic

To ensure the completeness of this experiment, we consider a scenario where the majority of workers are experts, as presented in Figure 4.6(a). We also consider a small sample of bad annotators just to have little noise in our population.

As expected, using a population with a majority of experts we achieve the best results: the average of accuracy was around 77 % in all crowdsourcing methods. As in section 4.3.2, crowdsourcing methods achieve the best results.



(a) Workers distribution



(b) Average of all models accuracy

Figure 4.6: Accuracy of crowdsourcing methods using a logistic distribution of worker's expertise

4.3.4 Gaussian Translated

This previous experiments had the goal of learning the behaviour of crowdsourcing methods. However, none of those experiment represents a realistic crowdsourcing population. For example, it is very unlikely to have workers with expertise inferior to 0.5. We can have this type of worker in two scenarios. The first is when the worker did not

understand the task and he is doing the opposite. For example, in a task defined by the instruction: "Click in the picture that does *not* contain a tree", where the worker did not read the *not*. A second scenario is when the worker is performing incorrectly deliberately. Moreover, the most popular crowdsourcing sites have some politics to prevent workers which do not have a minimum of expertise: gold questions are an example of these. For the described reasons, we will perform an experiment where the workers population follows the distribution presented in Figure 4.7(a), as we believe it is the closest to a average crowdsourcing population.

In turn, he Figure 4.7(b) represents the obtained results. As expected, the results were better than the previous experiments since we are using workers with an average expertise of 0.75. It is interesting to note that the standard deviation is almost null (the accuracy was almost the same for all crowdsourcing methods) and the judgements per element had no influence.

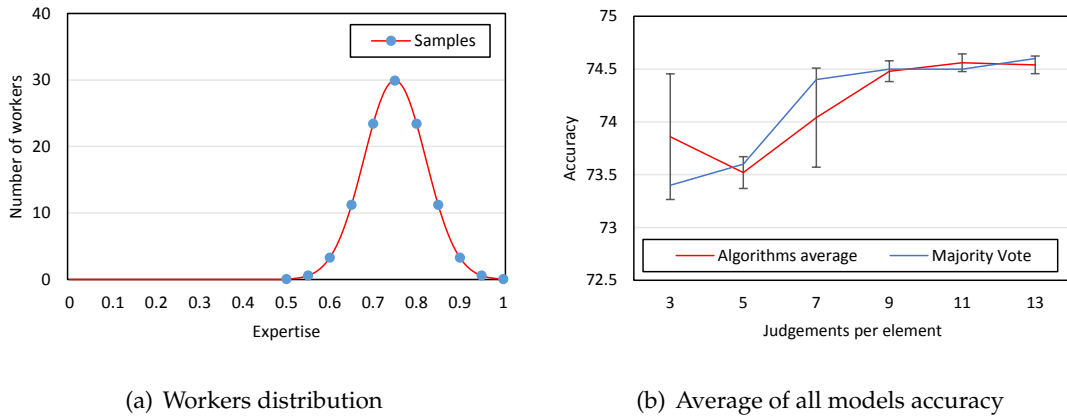


Figure 4.7: Accuracy of crowdsourcing methods using a Gaussian distribution translated of worker's expertise

4.3.5 Discussion

The experiments described previously show the crowdsourcing models behaviour as a whole. In this section we will analyse the individual performance of those methods against the majority vote. Table 4.1 presents the difference of each model accuracy against the majority vote. The column *MV* indicates de absolute accuracy of majority vote in each experiment. The relative accuracy of each crowdsourcing model against the majority vote is presented in its respective columns. The last row and column presents the average of each row and column, respectively.

As presented, there are more positive values than negative ones, which indicates that crowdsourcing models performed better in the majority of the experiments. Although we have some negative values, these are not significant since they are very close to zero. Moreover, DS achieved the major negative disparity of majority voting and it was only a difference of 0.3, meaning that majority vote classified correctly a total of 3 more elements

than DS. On the other hand, we can clearly see that crowdsourcing models were much better in experiments with the Inverse Gaussian and Logistic populations. It is interesting to notice that these experiments were the ones to consider a greater number of bad workers.

Experiments	MV	CUBAM	DS	GLAD	RY	ZC	Average
Constant	49.3	0.7	30.4	-0.1	0.7	0.1	6.4
Gaussian	48.5	0.1	-0.3	-0.1	-0.1	-0.1	-0.1
Inverse Gaussian	50.4	5.5	4.0	4.8	4.0	5.9	4.8
Logistic	76.4	3.3	3.2	2.3	1.2	1.1	2.2
Gaussian translated	74.4	0.0	0.0	0.1	0.0	0.0	0.0
Average	0.0	1.9	7.4	1.4	1.2	1.4	

Table 4.1: Average of relative accuracies when comparing against the majority voting.

4.4 Real Data Experiment

In the previous section we study the behaviour of crowdsourcing methods using synthetic data. However, we are interested in using those methods in a real scenario. Thus, in this section we will analyse the performance of crowdsourcing methods using a real crowdsourcing dataset.

4.4.1 Dataset

We will use a sample of 108 images from the bluebirds dataset. In this dataset, 29 crowdsourcing workers from mturk had to discriminate between two species of blue birds: Indigo Bunting and Blue Grosbeak. This resulted in a total of 472 labels, collected through non-expert annotators. Additionally, the dataset contains the ground-truth labels that were annotated by experts. An example of these two species can be seen in Figure 4.4.1.



(a) Indigo Bunting specie

(b) Blue Grosbeak specie

Figure 4.8: Species that each annotator has to identify

The evaluation of these six models will be performed through the computation of accuracy in each model for a various samples of annotators. With this approach, we aim to see each model's behaviour when the number of annotators increases and, consequently, the number of labels. On a second evaluation we will stress all the six models. This is achieved by adding noisy labels to our data sample. We consider two types of noise:

- **Random annotators** - Annotators which always produce random labels.
- **Adversarial annotators** - Annotators which always produce wrong labels.

4.4.2 Results

As explained, we run all the algorithms with different number of judgements per image, from 1 to 29 judgements. Since we have many crowdsourcing workers, we perform this experience 30 times, allowing us to use different workers in each run. Moreover, it also allows us to subtract some bias from some worker's expertises. From these 30 runs, we have calculated the middle value. One could use the average instead the middle value, however, the middle value represents a more realistic value since it actually is a result of using some real worker's labels. If we think in an algorithm which produce only accuracies of 0 or 100, the average is 50. However, in a real scenario that can not happen for any algorithm.

Figure 4.9 shows the accuracy for the different numbers of judgements per image. We can clearly observe that all models increase their accuracy when increasing the number of judgements. CUBAM and DS achieve the best results with almost 90 % of accuracy. Furthermore, while Raykar is the closest to these methods, the difference is still significant. The remaining two crowdsourcing methods, namely GLAD and ZC, can not even beat the majority vote. Is interesting to notice that all models converge around 11 judgements, meaning This means that paying for more than 11 judgements, meaning for this type of task is increasing the price without increasing the data quality.

Although CUBAM and DS are those who achieved better results, they performed worst for low judgements per image. This show us that we can not rely in a single crowdsourcing method: one must be aware of his data to select a crowdsourcing method. Overall, the obtained results suggest that we can use CUBAM and DS when we have many judgements per image. One the other hand, Raykar is the algorithm to use for low judgements per image.

4.5 Hybrid Experiment

As discussed in the previous section, the results of crowdsourcing models strongly depend on our data. Furthermore, the number of judgements per image in our dataset can determine the usage of a crowdsourcing model instead of other. However, it is not only the judgements per image that characterizes our data. Maybe the most important factor is

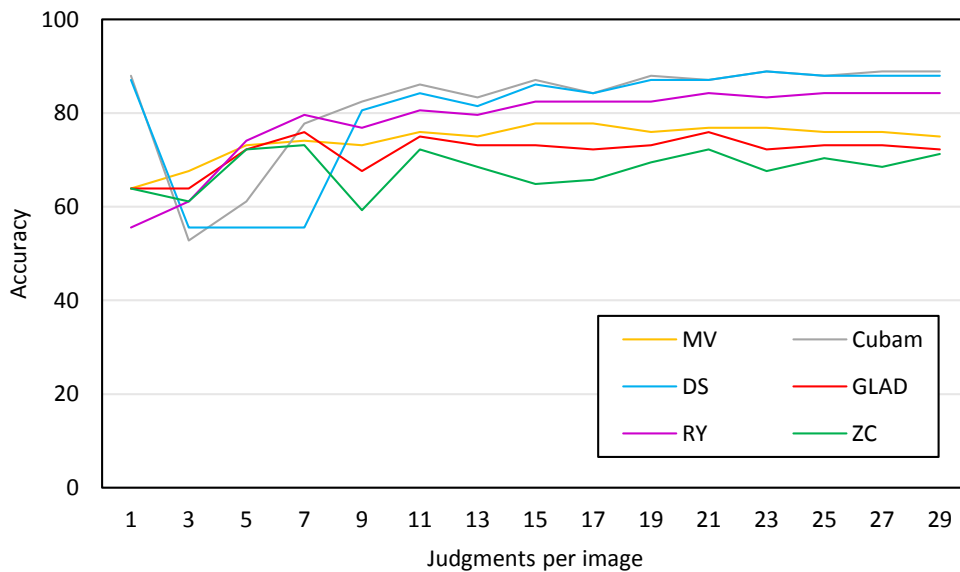
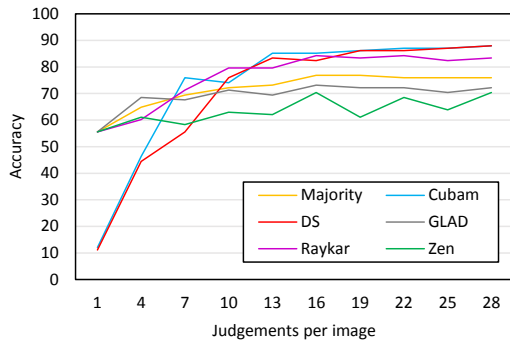


Figure 4.9: Accuracy of running crowdsourcing methods using bluebirds dataset.

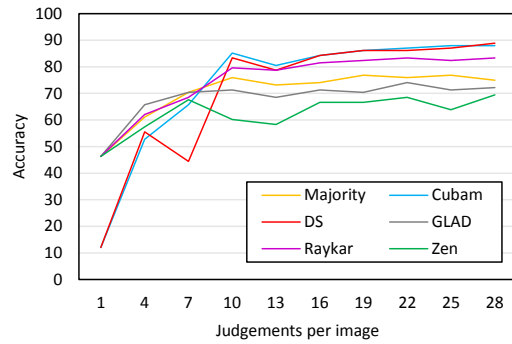
the workers' expertise. In section 4.3 we studied how different workers populations can affect the results. Moreover, we concluded that for some types of populations, the difference between crowdsourcing methods and the majority vote reveals to be statistically noticeable. However, those experiments were only possible because we used synthetic data, as it is really hard or even impossible to qualify the expertise of a real worker. One can even perform the same job in a crowdsourcing site many times but still does not know the type of population. We can, instead, explore an hybrid experiment where both real and synthetic data is used. Despite this experiment not allowing us to shape our population, we can change it and study how crowdsourcing models react to such change. Therefore, in the next two experiments, we will add noisy and adversarial workers to the bluebirds dataset. Noisy workers are those who randomly label an image. Adversarial annotators always choose the wrong label. By adding such workers to our dataset, we can simulate a scenario with characteristics that emphasize the problem which the models are willing to solve.

4.5.1 Random Workers

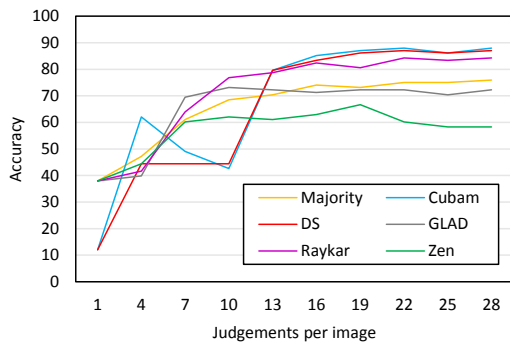
In the first experiment, we will add random workers to the bluebirds dataset. Noisy workers have 0.5 of expertise, meaning that they have the same probability to label an image correctly and incorrectly. Unlike the experiments made in Section 4.3, the images from the bluebirds dataset do not have an associated difficulty, which means that the label from a random worker is random. We will perform 6 experiments, adding from 1 to 6 random workers. Note that our random workers are the first to label our dataset. For



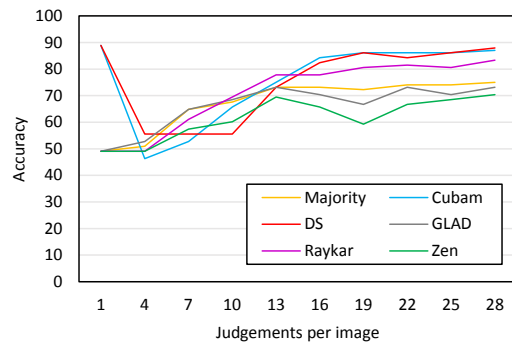
(a) Error rate with 1 noisy annotators



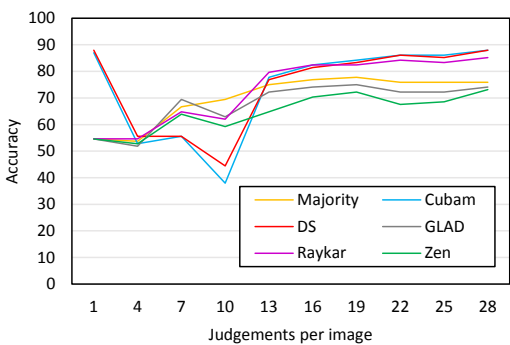
(b) Error rate with 2 noisy annotators



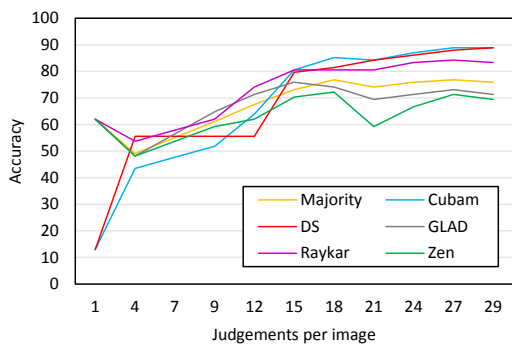
(c) Error rate with 3 noisy annotators



(d) Error rate with 4 noisy annotators



(e) Error rate with 5 noisy annotators



(f) Error rate with 6 noisy annotators

Figure 4.10: Accuracy of running crowdsourcing methods using bluebirds dataset with random workers.

example: if we add 1 random worker and request 1 judgement per image, all workers judgements will be random. As explained in Section 4.2, every worker contributes with same number of judgements. However, this is not true for random workers. A random workers have priority to label an image. This strategy allows us to analyse the real impact of these workers in our dataset.

Figure 4.10 presents the accuracy of running crowdsourcing methods for the described experiments. In a global analysis, we can see that for 1 judgement per image, almost all crowdsourcing methods achieves the same accuracy. This is expected when using only one worker: the estimated labels will be the labels of that worker. Moreover, all of those crowdsourcing methods achieves around 50 % of accuracy for 1 judgement which is a consequence of using at least one random worker since, in average, he will label correctly fifty percent of the times. The exception in this trend is CUBAM and DS: both start with high or low accuracy for 1 judgements.

As shown in Figure 4.10, by increasing the number of random annotators all crowdsourcing models will eventually achieve the same accuracy as in Section 4.4.2 (around 90 %). However, we observe that the number of judgements necessary to achieve that value increases with the number of random workers added: with more random workers, more judgements per image are needed for crowdsourcing methods to converge. In Section 4.4.2 we concluded that almost all algorithms converge with 11 judgements per image. Thus, one could think that adding n random workers the convergence happen for $11 + n$ judgements per image. However, we observe that some algorithms need less or the same number of judgements to converge. Despite this early convergence, they converge for a lower value of accuracy. On the other hand, CUBAM and DS need more than $11 + n$ judgements but they achieve almost the same accuracy as in the original bluebirds dataset. For example, when adding 5 random workers (Figure 4.10(e)) CUBAM and DS converges around 22 judgements per image.

4.5.2 Adversarial Workers

In the previous experiment we study the behaviour of crowdsourcing models when our dataset has random workers. Despite their random judgements, they will label correctly half of the times. Now we will study the behaviour of crowdsourcing models when the dataset has workers which never label correctly: adversarial annotators. This type of workers is not common to occur in a real scenario. We identify two scenarios to have this kind of workers in our crowdsourcing dataset: (1) The worker is purposely labeling incorrectly; (2) The worker did not understand the instructions and is doing the opposite of what is requested. The setup of this experiment is similar to the one presented in the previous experiment (Section 4.5.1). This experiment will add more noise to the dataset. Therefore, we will only perform experiments from 1 to 4 adversarial workers.

The results presented in Figure 4.5.2 are very different from the previous experiments. When considering 1 judgement, the accuracy is 0 or very close to 0 in all models. This

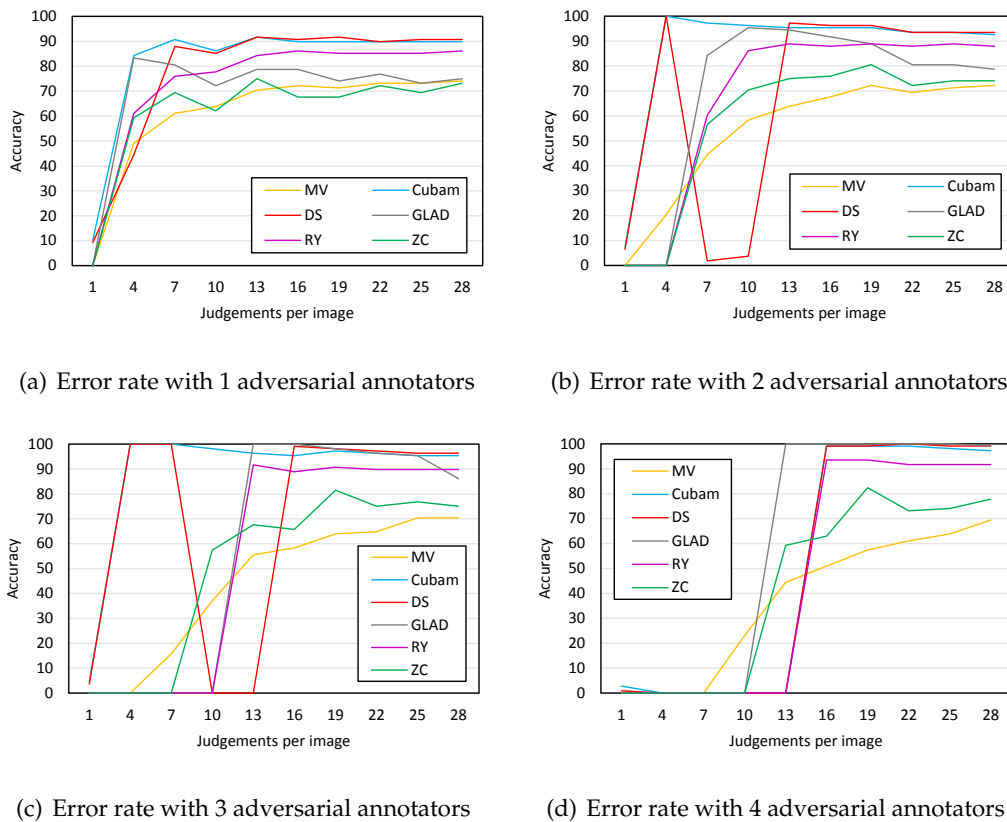


Figure 4.11: Accuracy of running crowdsourcing methods using bluebirds dataset with adversarial workers.

happens because that adversarial worker labeled all images incorrectly. The same happens for 2,3 and 4 judgements, per image when the same amount of adversarial workers is used. We can clearly see that, in all 4 experiments the convergence is not as smooth as in the random experiment. When the convergence process occurs, it occurs faster than in the above scenarios. Also, this convergence happens sooner than in the random experiment and even when using only the original dataset. We can observe in Figure 4.11(a) that we only need 7 judgements per image to achieve an accuracy of 90 percent. Using 2 adversarial workers (Figure 4.11(b)) we only need 4 judgements to CUBAM achieves 100 percent of accuracy. This results seem contradictory due the nature of the used workers. However, the results suggest that crowdsourcing methods can use adversarial judgements to infer the true label. In a binary experiment as such this one, identifying the adversarial label allow us, by elimination, to infer the true label.

4.6 Summary

In this chapter we studied the state-of-the-art crowdsourcing methods. In Section 4.1, we described each one of those methods in and the main objectives of the authors. In Section 4.3 we study how different workers population affect the crowdsourcing methods

output. We observe that for some types of population using such methods is imperative. Whereas for other types of population the gain is not so significant. However, overall crowdsourcing methods outperformed the traditional majority voting.

In Section 4.4 we perform an experiment using a real crowdsourcing dataset: bluebirds. This allows us to comprehend the behaviour of crowdsourcing models in a real scenario. It was evident that CUBAM and DS were superior when using a high value of judgements. However, for a low number of judgements per image Raykar can perform better. Moreover, these methods were the only ones to outperform the majority vote. GLAD and DS do not beat the majority voting in this experiment.

Further, in Section 4.5 we perform a hybrid experiment where we use the bluebirds dataset and some non-expert synthetic workers. This experiment allowed us to understand how crowdsourcing methods will react in the presence of two types of bad workers: random and adversarial workers. We conclude that random workers will require more judgements per image until they reach convergence. Whereas adversarial annotators make the convergence happen sooner and faster. Despite the lack of expertise of these workers, they can be an advantage to crowdsourcing methods to, by elimination, infer the true labels.

In the experiments aforementioned, we clearly observed that CUBAM and DS are the ones who produce the best results. However, sometimes these methods require more judgements to achieve such results. The use of one crowdsourcing method strongly depends on the objective in hand. If we want to create a cheap dataset where losing some accuracy is not so important, one may collect few judgements per image and use Raykar. On the other hand, if the objective is to create the most reliable dataset the choice is to collect a large number of judgements per image and use CUBAM. Also, note that the number of judgements is proportional to the difficulty of the task. We consider that labelling the bluebirds dataset is a task with medium difficulty for an average person. Easier tasks will require less judgements while harder tasks will require more judgements.



Learning Classifiers with Weak-labels

5.1 Methodology

In a real scenario, we want a crowdsourcing model which can generate labels as close as possible to the ground-truth, allowing us to train classifiers with a high accuracy. In the previous chapters, we observed that crowdsourcing can be a good alternative to the ground-truth when labelling a dataset. In this chapter, we will study the reliability of crowdsourcing labels in training a facial expressions classifier. In order to perform this experiment, we will use two datasets: the Cohn and Kanade dataset and the Novaemotions dataset. In the first one, facial expressions were collected in a controlled environment where subjects were aware of the aim of the experiment. In the second one, facial expressions were collected while players interacted affectively with the game. The process to collect the crowdsourcing labels is explained in Chapter 3. These datasets will be used to train three classifiers: (1) the k -NN classifier, (2) the weighted k -NN, and (3) the Kernel Density Estimation (KDE). The three classifiers were implemented in the context of this thesis to allow the use of weak-labels.

The features used in these experiments were extracted with a bank of Gabor Filters, since this is widely used in facial expression recognition problems. The Gabor filter detects the contours of an image at different scales and orientations. For that reason, it is a good way to detect traits that differentiates the facial expressions, for example the shape and eyebrows's and mouth's orientation.

5.1.1 k -Nearest Neighbors

In a k -NN classifier, we have a training set, which contains, for each training element, a multidimensional feature space and its corresponding label. For a test query, the euclidean distance is computed against all the elements on the training set and only the k nearest elements are returned. To determine the class of the query instance we can simply calculate the majority of the classes on the set of the k elements. This classifier allows us to better understand our problem and, more importantly, find which aspects can improve the use of crowdsourcing labels in a supervised classification.

Weighted k -NN. The k -NN classifier suffers from the same problem as the majority voting method: every k -neighbour has equal contribution when classifying a new image. Therefore, a natural extension of the k -NN is to assign different weights to each neighbour. A simple extension is to use the distance to the neighbour as the weight, and this allow us to give more importance to the nearest neighbours and less to the distant ones. We will refer to this modified nearest neighbour classifier as weighted k -NN. The contribution of each nearest neighbour is the inverse of the distance ($1/d$) to the test sample.

5.1.2 Kernel density estimation

The KDE is an approach to estimate the probability density function of the training data. It applies a weighting function, or Kernel, to every points in our training set. This approach differs from the k -NN, since the last one uses a small fraction (k elements) of the entire set to classify a new element. The KDE uses all elements of our training set to estimate the density of the distribution.

The contribution of each element also differs from the weighted k -NN: instead of using the Euclidean distance to weight each neighbor, KDE uses a probability distribution function to compute the contribution of every training sample. The functions defining the contribution of a data sample, are the so called Kernel function. Popular Kernel functions include the Gaussian distribution, the Epanetchnikov distribution and the Laplacian distribution. For a matter of convenience, we will use the Gaussian function:

$$K(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2}$$

Now that we defined the Kernel function, we are interested in estimating a function $f(x)$ that represents the aggregated contributions of all training samples in order to estimate the label of an new element. The KDE allow us to estimate the density function on a given point x , as the sum over all training samples:

$$\hat{f}(x) = \frac{1}{n} \sum_{i=1}^n K_h(x - x_i)$$

Due to the fact that $f(x)$ is dependent on the distance of point x to the training samples, we need to compute this sum for every test image that we need to classify.

Formally, we will have one function $f_l(x)$ for each label l of our problem. In our case, we will have a function $f(x)$ for each facial expression that we are considering. Therefore, we need to define the weight that each training element has in each facial expression. One approach is to assume that each training sample contributes for the density function of its own label, consequently, the contribution for the remaining density functions is zero. With this assumption we extend the previous definition as follows:

$$\hat{f}_l(x) = \frac{1}{n} \sum_{i=1}^n K_h(x - x_i) * \mathbf{1}_l(x_i)$$

where $\mathbf{1}_l(x_i)$ is an indicator function, taking the value 1 if the sample x_i belongs to the class l and 0 otherwise. In the presence of weak-labels, a facial expression can be annotated with a several labels, for example, a mix of neutral and happy. These assumptions allow us to extend the $f(x)$ to support weak-labels:

$$\hat{f}_l(x_0) = \frac{1}{n} \sum_{i=1}^n K_h(x_0 - x_i) * p(L_i = l | x_i)$$

where $p(L_i = l | x_i)$ is given by the crowdsourcing annotations.

In a multi-class problem, a common challenge is the imbalance of number of training samples of each class. We have two ways of solving this problem: one approach is to transform each density function in one probabilistic function, which is achieved by ensuring that the area beneath the curve is one; another approach is to establish a prior π of each class j , and for doing that we will assume that these priors are the proportion of each class in our training set. This last approach allows us to use the Bayes' theorem to classify a new element:

$$p(C = c_0 | X = x_0) = \frac{\pi_{c_0} f_{c_0}(x_0)}{\sum_{c=1}^C \pi_c f_c(x_0)}$$

This definition allow us to compute the probability of one image x_0 belonging to a certain class c_0 . To classify new images, we only need to find which class has the highest probability or, in other words, the class c that maximizes the above expression.

5.2 Datasets

In the previous chapter, we analysed the behaviour of each model against a known ground-truth. The experiments presented in that chapter, the Bluebirds experiment, do not represent our aim entirely. For that reason, we will perform an experiment by using two facial expressions dataset: the Cohn-Kanade extended (CK+) dataset, and the Novaemotions dataset.

5.2.1 Cohn-Kanade

The CK+ (Figure 5.1) dataset contains 593 sequences of images from 123 subjects, where each sequence includes several images showing the evolution between a neutral facial expression to another one. The majority of these sequences are annotated by experts which give us the ground-truth. Only the first and the last facial expression of each sequence are annotated by experts.

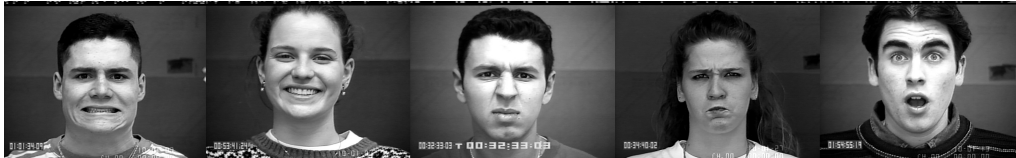


Figure 5.1: Example of CK+ dataset.

5.2.2 Novaemotions

The Novaemotions (Figure 5.2) dataset is a facial expression dataset in which they were captured while players interacted with a game. Therefore, these facial expressions were not captured in ideal situations, such as the face orientation or the lighting. Due to the nature of this dataset, some people were so engaged with the game that they were laughing or talking while performing the requested facial expression, thus, the difficulty is much harder than in the CK+ dataset. In total, more than 40,000 images were captured.

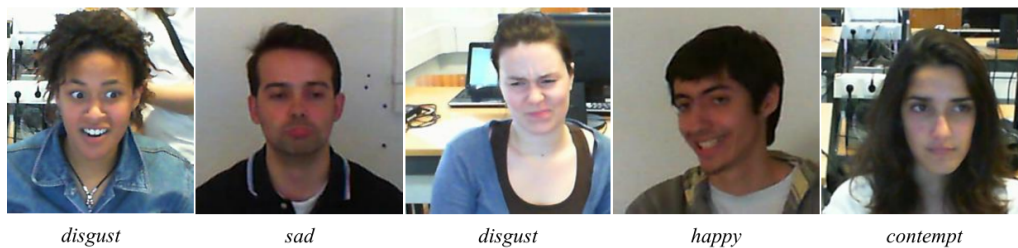


Figure 5.2: Example of Novaemotions dataset.

5.2.3 Crowdsourcing labels

The process used to collect the crowdsourcing labels for the CK+ dataset was similar to the one used for the Novaemotions dataset described in Chapter 3. Thus, we collected 5 labels per image and we were able to measure the accuracy of crowdsourcing methods against the known ground-truth, using a multiclass facial expressions dataset. For the crowdsourcing methods that can not perform a multiclass classification, we followed a binary classification for each facial expression. For example, to estimate which images belong to the class *happy*, the votes for this class are considered 1, whereas all the other votes are considered 0. At the end, we merged all the results for all facial expressions.

Sometimes, crowdsourcing methods could not infer any label for an image. Therefore, we only considered images where all the methods estimated one label per image. In total, we have 1081 images from the CK+ dataset, which fulfills these requirements.

Facial expression	MV	CUBAM	DS	GLAD	RY	ZC	Total
Angry	80.00	80.00	78.75	76.25	80.00	81.25	80
Disgust	96.81	72.34	98.94	95.74	96.81	94.68	94
Fear	97.22	30.56	94.44	97.22	97.22	77.78	36
Happy	91.89	88.51	91.22	91.22	91.22	93.24	148
Neutral	88.16	89.02	84.73	87.14	90.05	90.05	583
Sad	93.88	69.39	89.80	91.84	89.80	91.84	49
Surprise	100.00	74.13	100.00	100.00	100.00	98.60	143
Accuracy	90.74	82.28	88.71	89.68	91.45	91.01	1133

Table 5.1: Precision of each model labels against known groundtruth

The results of Table 5.2.3 show the accuracy of each crowdsourcing method for each facial expression. The last line presents the accuracy of each crowdsourcing method, whereas the last column present the number of images of each facial expression. Note that, we did not take into account the facial expression *contempt* because we had only one example. In the multiclass problem, the best accuracy was 91.45 % and that result was achieved by RY. It is interesting to notice that RY only support binary labels but achieved better results than DS and ZC, which actually support multiclass. On the other hand, CUBAM achieved only 82.28 %. We identified two causes for this: (1) our approach to use CUBAM in a multiclass problem did not work, and (2) CUBAM can not handle a dataset with different class proportions. However, these results show that we can rely in crowdsourcing to collect facial expressions labels.

5.3 Classifying Facial Expressions with Weak Labels

In this section we will evaluate the use of crowdsourcing labels to train a classifier. We will use the classifiers presented in Section 5.1: k -NN; weighted k -NN and KDE. Also, we will train these classifiers with the ground-truth.

5.3.1 Training and Test Data

For this experiment, each classifier was trained with two distinct datasets: the CK+ and Novaemotions datasets. Although both are facial expression datasets, it is important to note that the facial expressions presented in CK+ were captured in an environment under the complete control of the authors and the subjects were aware of the final objective of the authors. Unlike CK+, the facial expressions from the Novaemotions dataset were captured while players were engaged in a video game. To build these datasets we did

not control the environment neither were the subjects aware of the true purpose behind the game.

In order to have a dataset with a ground-truth to train and test, we split the CK+ dataset in two subsets: CK-1 and CK-2. To perform this division, we ensured that images from the same sequence belonged to the same subset and had a ground-truth label. This approach avoided training and testing with facial expressions from the same person. Additionally, we ensured that both subsets had approximately the same amount of training examples per class. Afterwards, we trained all classifiers with one of these subsets using labels produced by crowdsourcing methods and the ground-truth. The remaining subset is used to test.

The previous split only used images from the ground-truth. Therefore, many images are not used. To address this problem, we performed another division. We created a subset of CK+ dataset which contained only images from the middle of each sequence (CK-M) and another subset which only contained the first and the last images of each sequence (CK-FL). This allowed us to train with CK-M and test it with CK-FL but not the opposite, because only the CK-FL has a ground-truth.

The previous experiment allowed us to predict what to expect when training with data not affected by conditions such as lighting, background noise and pose. However, our objective was to use facial expressions captured while players interact with a game. Therefore, we performed an experiment similar to the one above, where we trained the KNN classifier using the Novaemotions dataset, which fulfilled our requirements. The process used to collect the labels was also similar to the CK+ dataset. However, we did not have a ground-truth that allowed us to test the classifier with same kind of data that was used to train it. Thus, we will use the images with ground-truth from the CK+ dataset.

5.3.2 Results

The results in Table 5.2 show the accuracies when training the k -NN, weighted k -NN and KDE classifiers with images from the subsets presented in Section 5.3.1 and labels from different crowdsourcing methods. Also, whenever it was possible, the ground-truth that was used to train the classifier. At first glance, we can observe that training with CK-M and testing with CK-FL achieves the best results with accuracies around 85%. The reason for these high results is because we did not avoid including images from the same person in training and testing. Additionally, this approach allowed us to understand if workers and the crowdsourcing methods could infer the correct label for the images in the middle of a sequence. These are harder to identify than the first one or the last one, because the middle facial expressions are the transition between a neutral facial expression state and a well defined facial expression. A perfect worker should be capable to label a middle facial expression as *neutral* or its final state. This assumption proved that workers, in fact, labelled correctly the middle images. However, CUBAM did not ensure the same quality

Classifier	Train	Test	MV	Glad	CUBAM	ZC	RY	DS	GT
KNN	CK-M	CK-FL	85.53	85.53	78.95	84.82	85.32	85.32	—
KNN-W	CK-M	CK-FL	85.53	85.53	79.96	85.32	85.63	84.82	—
KDE	CK-M	CK-FL	85.73	85.63	79.45	83.40	85.43	86.03	—
KNN	CK-1	CK-2	60.29	60.29	59.39	59.93	60.29	60.82	60.82
KNN-W	CK-1	CK-2	60.64	60.64	59.57	60.11	60.64	61.18	60.47
KDE	CK-1	CK-2	58.14	58.14	57.60	57.25	58.14	57.78	57.25
KNN	CK-2	CK-1	60.84	60.61	58.97	61.07	60.61	60.14	61.54
KNN-W	CK-2	CK-1	60.84	60.61	59.44	60.37	60.61	60.37	61.54
KDE	CK-2	CK-1	57.34	57.34	56.64	57.58	57.34	57.11	58.51
KNN	Nova	CK-FL	31.98	28.85	31.98	33.30	31.88	31.28	—
KNN-W	Nova	CK-FL	26.72	26.01	25.91	29.55	28.95	26.21	—
KDE	Nova	CK-FL	24.90	24.29	24.39	26.82	26.62	23.99	—

Table 5.2: Accuracies of training various classifiers (k -NN, weighted k -NN and KDE) using crowdsourcing labels from different methods.

as the other crowdsourcing methods with a difference of around 6%. Comparing the classifiers, we could not pin-point a single classifier that was better across all experiments.

Analysing the second and third set of experiments, with CK-1 and CK-2 subsets, the results are almost the same when training with one subset or the other. Unlike the previous experiment, we can clearly observe that k -NN performed better than the other classifiers, although the difference is not as significant as when comparing with the weighted k -NN. On the other hand, when comparing it with KDE, the difference is around 2%. CUBAM was not capable of performing as well as the other crowdsourcing methods. In this experiment we trained the classifiers with the ground-truth. Unexpectedly, training the weighted k -NN and KDE classifiers with the crowdsourcing labels achieved better results than with the ground-truth. In the former classifier, the DS method was the best with an accuracy of 61.18% while in KDE three models achieved 58.14%: MV, Glad and RY. In the remaining four experiments, although the ground-truth achieved the best results, the difference to the crowdsourcing methods was as little as 1%. These results suggests that crowdsourcing labels can actually be used as an alternative to the ground-truth.

The fourth and last set of experiments involved training each classifier with the Novaemotions dataset and testing it with CK+ dataset. As explained in Section 5.2, the images in the CK+ dataset were collected in a controlled environment where the authors could control aspects like lighting, background as well as the subjects pose. The images in Novaemotions dataset were collected in a real environment while players affectively interacted with a game. These differences explains the results obtained and presented in Table 5.2 where the best result was 33.30%. It is interesting to note that, in previous experiments, ZC never surpassed the other crowdsourcing methods, yet in this experiment achieved the best results with a significant margin. This suggests that ZC can perform better with a noisier dataset than the other crowdsourcing methods.

5.4 Summary

In this chapter we analysed the performance of three classifiers in the presence of weakly labeled training data. Section 5.1, described each classifier and how we can use them in our context. The k -NN is the simplest classifier used where the label assigned to the new image is given by the majority class of the k closest neighbours. The weighted k -NN allowed us to give different weights to each neighbour (i.e. the weak-labels are included as probabilities and not as binary labels). Similarly, the training elements in the KDE have different contributions to classify a new image. This contribution is given by the Gaussian Kernel used. We conclude that using different weights per training element can be a solution to use crowdsourcing labels to train a classifier.

In Section 5.2 we presented two distinct facial expressions datasets: CK+ and Novaemotions. The first is a facial expression dataset where the images were collected in a controlled environment, whereas the images of Novaemotions dataset were collected in an uncontrolled environment. In this chapter we also compared the crowdsourcing methods output against the ground-truth of the CK+ dataset. We observed that crowdsourcing labels were very similar to the ground-truth with an accuracy of 94% in three crowdsourcing methods: MV, GLAD and RY. These results suggested that crowdsourcing output is actually a viable alternative to more expensive ground-truth.

In the last section (5.3) we studied the use of crowdsourcing labels to train a classifier. We divided the CK+ dataset in two subsets: one with the intermediate images of each sequence (CK-M) and another with the first (neutral) and the last image (CK-FL). Also, we divided the CK+ in two more subsets ensuring that images from the same sequence were in the same subset (CK-1 and CK-2). These subsets allowed us to perform four experiments: (1) train the classifiers with CK-M and test it with CK-FL; (2) train the classifiers with CK-2 and test it with CK-1; (3) train the classifiers with CK-1 and test it with CK-2; (4) train the classifiers with Novaemotions and test it with CK-FL. The first experiment achieved the best results, which leads us to believe that crowdsourcing workers could actually label the most ambiguous images from the CK+ dataset correctly. The worst results were achieved in the fourth experiment. The reason for this is that the nature of both datasets is quite different. Despite these worst results, ZC stands above the other crowdsourcing methods in this experiment. This suggests that ZC is better when we use a noisier crowdsourcing dataset to train a classifier.



Conclusion

This thesis explored the use of crowdsourcing to label a large-scale facial expression dataset. Unlike professional or experts, we can not consider the data produced by a crowdsourcing worker as a ground-truth. Therefore, it is imperative to collect many judgements per image. However, this approach force us to use methods that merge these labels into a single label, ideally equal to the ground-truth. The ease of collecting crowdsourcing data in combination with their low costs allow us to consider strategies to use crowdsourcing data as a reliable and cheap alternative to the ground-truth given by experts. In sum, during this thesis we tried to answer the following questions:

- Q1) How to collect facial expressions labels using only crowdsourcing?
- Q2) How to merge labels for the same image that were given by different workers?
- Q3) How effective is crowdsourcing data in training a classifier?

Concerning the crowdsourcing process to obtain the facial expressions labels (Q1), we started by identify factors that directly or indirectly influence the data collected from crowdsourcing sites. A crowdsourcing job has several micro-tasks where workers request pages of micro-tasks to perform (not the whole job). We concluded that the number of micro-tasks presented in crowdsourcing pages may directly influence the quality of the data. Once the worker is rewarded at the end of each page, pages with many micro-tasks will prevent that worker from leaving the task at any time. To reinforce this idea, using fewer micro-tasks per page makes the data quality increase over time. In other words, workers which perform more micro-tasks at their "own will" produce more reliable data. However, the everlasting question in crowdsourcing is how much to pay to each worker. In this thesis, we observed that increasing the price also increases the data quality but

this increase is not significant. This is a subtle question, but we also believed that this will be essential in the future of crowdsourcing. After setting the optimal parameters for a crowdsourcing job, we collected 5 judgements for 40,982 facial expressions, which resulted in the creation of the Novaemotions dataset. A first version of this work was published in Workshop CrowdMM'13, co-located with ACM Multimedia 2013 [5].

Regarding the merging of the labels given by different workers (**Q2**), we performed an evaluation of the state-of-the-art of several crowdsourcing methods: Dawid and Skene (DS) [40], GLAD [41], CUBAM [42], Raykar [43], Zen [44], and the baseline Majority voting. These methods aim to infer the true label among various crowdsourcing labels for the same image. Unlike the majority voting, these methods model other attributes, such as worker's expertise, bias and task difficulty. To perform this evaluation, we created synthetic workers to be able to design different worker's population. This allowed us to understand the behaviour of crowdsourcing methods with different population of worker's expertise. We concluded that the crowdsourcing methods can perform better than the traditional majority voting. However, the gain is not so significant for some types of workers population. Despite that, when the population includes good workers the gain is between 3 and 5 percent when comparing with the majority voting method.

With respect to crowdsourcing being an alternative to the ground-truth (**Q3**) we also tested the crowdsourcing methods using a real dataset with a known ground-truth. We verified that CUBAM, DS, and Raykar outperformed majority voting. Additionally, we added two types of noisy workers in this dataset: random workers and adversarial workers. Random workers delayed the convergence to a stable label (need more judgements per image), but achieved the same accuracy; adversarial workers made the convergence happened sooner and achieved even better results than when we used the clean dataset. Moreover, the majority voting achieved the worst results when used adversarial annotators.

In conclusion, we believe that the choice between one of these methods depends on the problem at hand. While CUBAM and DS are the best methods to achieve the best results, but they need more judgements per image to achieve such results. Raykar achieved almost the same accuracy with less judgements per image. Therefore, to create the most reliable data we should request a considerable number of judgements and use the CUBAM method to estimate the true labels. On the other hand, if we want to build the best price-quality ratio dataset, we should request few judgements and use the Raykar method.

We also carried out an evaluation (**Q3**) to test if the crowdsourcing labels can be replace the ground-truth given by experts. It was clear that the data collected from crowdsourcing are identically to the one produced by experts (more than 90% of accuracy). Moreover, we trained three classifiers using crowdsourcing labels. The obtained results revealed that training with this type of data achieved almost the same results as training with the ground-truth.

6.1 Future work

This thesis studied the reliability of using crowdsourcing as a source of knowledge. However, this recent research field is in its early stages – we foresee some of the crowdsourcing future as the expansion of the following research fields:

- **Human-in-the-loop strategies** - The evolution of machine learning algorithms allows us to use computers to solve several difficult problems, such as facial expression recognition. Nevertheless, harder tasks are still a problem. In this case, crowdsourcing can help to disambiguate some computer's results or even validate them. In a near future, it will be expected that the symbiosis between humans and computers will increase (as is the case in search engine results quality control).
- **Reward strategies vs *pay-per-task* approach** - The reliability of crowdsourcing data is strongly connected to workers' commitment. This means that crowdsourcing methods are worthless if crowdsourcing data is unreliable. We can see this process as a house of cards where we need a solid base to keep the structure steady. Therefore, we can perceive the emergence of new approaches to maximize the workers' commitment and, consequently, the data quality. Since money is the most important incentive in crowdsourcing sites, some approaches replace the traditional *pay-per-task* for other strategies, such as *Winner-Takes-It-All*, where the most successful worker wins all the money involved [72].
- **Multi-class crowdsourcing methods** - The state-of-the-art crowdsourcing methods do not address multi-class problems explicitly. Although some of these methods formalize a multi-class approach, the majority of these works did not perform a proper evaluation. Our approach to make these methods handle a multi-class classification was not the ideal one. This field requires a more detailed study and evaluation in the future.
- **Learning a classifier and the ground-truth jointly** - In this thesis, we evaluated the crowdsourcing methods and the use of weak-labels to train a classifier individually. This process made us lose valuable information. For example: when we trained a classifier with crowdsourcing methods labels we lost the individual worker's responses. A more comprehensive approach would be capable of jointly learning a classifier while it estimates the true labels. Nowadays, Raykar et al. [43] was the only one to propose such approach.

Bibliography

- [1] A. Mourão and J. Magalhães. “Competitive affective gaming: Winning with a Smile”. In: *Proceedings of the 21st ACM international conference on Multimedia*. ACM. 2013, pp. 83–92.
- [2] A. Mourão, P. Borges, N. Correia, and J. Magalhães. “Facial Expression Recognition by Sparse Reconstruction with Robust Features”. In: *Image Analysis and Recognition*. Springer, 2013, pp. 107–115.
- [3] P. Lucey, J. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews. “The Extended Cohn-Kanade Dataset (CK+): A complete dataset for action unit and emotion-specified expression”. In: *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*. 2010, pp. 94–101. DOI: 10.1109/CVPRW.2010.5543262.
- [4] L. Yin, X. Chen, Y. Sun, T. Worm, and M. Reale. “A high-resolution 3D dynamic facial expression database”. In: *Automatic Face Gesture Recognition, 2008. FG '08. 8th IEEE International Conference on*. 2008, pp. 1–6. DOI: 10.1109/AFGR.2008.4813324.
- [5] G. Tavares, A. Mourão, and J. Magalhaes. “Crowdsourcing for affective-interaction in computer games”. In: *Proceedings of the 2nd ACM international workshop on Crowdsourcing for multimedia*. ACM. 2013, pp. 7–12.
- [6] A. J. Quinn and B. B. Bederson. “Human Computation : A Survey and Taxonomy of a Growing Field”. In: (2011), pp. 1403–1412.
- [7] M.-C. Yuen, L.-J. Chen, and I. King. “A survey of human computation systems”. In: *Computational Science and Engineering, 2009. CSE'09. International Conference on*. Vol. 4. IEEE. 2009, pp. 723–728.
- [8] J. Howe. “The rise of crowdsourcing”. In: *Wired magazine* 14.6 (2006), pp. 1–4.
- [9] J. Howe. *Crowdsourcing: A Definition*. <http://crowdsourcing.typepad.com/>.

- [10] A. J. Quinn and B. B. Bederson. “Human computation: a survey and taxonomy of a growing field”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM. 2011, pp. 1403–1412.
- [11] S. Nowak and S. Rüger. “How reliable are annotations via crowdsourcing: a study about inter-annotator agreement for multi-label image annotation”. In: *Proceedings of the international conference on Multimedia information retrieval*. MIR '10. Philadelphia, Pennsylvania, USA: ACM, 2010, pp. 557–566. ISBN: 978-1-60558-815-5. DOI: 10.1145/1743384.1743478. URL: <http://doi.acm.org/10.1145/1743384.1743478>.
- [12] R. Snow, B. O’Connor, D. Jurafsky, and A. Y. Ng. “Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks”. In: *Proceedings of the conference on empirical methods in natural language processing*. Association for Computational Linguistics. 2008, pp. 254–263.
- [13] C. Wah. “Crowdsourcing and its applications in computer vision”. In: ().
- [14] G. Zoo. *Galaxy Zoo*. <http://www.galaxyzoo.org/>.
- [15] J. Lee, W. Kladwang, M. Lee, D. Cantu, M. Azizyan, H. Kim, A. Limpaecher, S. Yoon, A. Treuille, and R. Das. “RNA design rules from a massive open laboratory”. In: *Proceedings of the National Academy of Sciences* 111.6 (2014), pp. 2122–2127.
- [16] L. Von Ahn. “Games with a purpose”. In: *Computer* 39.6 (2006), pp. 92–94.
- [17] L. Von Ahn and L. Dabbish. “Labeling images with a computer game”. In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM. 2004, pp. 319–326.
- [18] Amazon. <https://www.mturk.com/>.
- [19] Crowdfunder. <http://www.crowdfunder.com/>.
- [20] M.-C. Yuen, I. King, and K.-S. Leung. “A Survey of Crowdsourcing Systems”. In: *Privacy, security, risk and trust (passat), 2011 ieee third international conference on and 2011 ieee third international conference on social computing (socialcom)*. 2011, pp. 766–773. DOI: 10.1109/PASSAT/SocialCom.2011.203.
- [21] J. Moehrmann and G. Heidemann. “Efficient Annotation of Image Data Sets for Computer Vision Applications”. In: *Proceedings of the 1st International Workshop on Visual Interfaces for Ground Truth Collection in Computer Vision Applications*. VIGTA '12. Capri, Italy: ACM, 2012, 2:1–2:6. ISBN: 978-1-4503-1405-3. DOI: 10.1145/2304496.2304498. URL: <http://doi.acm.org/10.1145/2304496.2304498>.
- [22] A. Sorokin and D. Forsyth. “Utility data annotation with amazon mechanical turk”. In: *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on*. IEEE. 2008, pp. 1–8.
- [23] *Street Bump*. <http://www.streetbump.org/>.

- [24] S. Deterding, M. Sicart, L. Nacke, K. O'Hara, and D. Dixon. "Gamification. using game-design elements in non-gaming contexts". In: *Proceedings of the 2011 annual conference extended abstracts on Human factors in computing systems - CHI EA '11* (2011), p. 2425. DOI: 10.1145/1979742.1979575. URL: <http://portal.acm.org/citation.cfm?doid=1979742.1979575>.
- [25] S. Deterding and D. Dixon. "From Game Design Elements to Gamefulness : Defining " Gamification "" ". In: (2011), pp. 9–15.
- [26] S. Ahern, M. Davis, D. Eckles, S. King, M. Naaman, R. Nair, M Spasojevic, and J Yang. "Zonetag: Designing context-aware mobile media capture to increase participation". In: *Proceedings of the Pervasive Image Capture and Sharing, 8th Int. Conf. on Ubiquitous Computing, California*. 2006.
- [27] L. Von Ahn, B. Maurer, C. McMillen, D. Abraham, and M. Blum. "recaptcha: Human-based character recognition via web security measures". In: *Science* 321.5895 (2008), pp. 1465–1468.
- [28] <http://eterna.cmu.edu/web/>. <http://eterna.cmu.edu/web/>.
- [29] A. M. Koblin. *The sheep market*. <http://www.thesheepmarket.com/>.
- [30] R. E. Jack, O. G. B. Garrod, H. Yu, R. Caldara, and P. G. Schyns. "Facial expressions of emotion are not culturally universal". In: *Proceedings of the National Academy of Sciences* 109.19 (2012), pp. 7241–7244. DOI: 10.1073/pnas.1200155109. eprint: <http://www.pnas.org/content/109/19/7241.full.pdf+html>. URL: <http://www.pnas.org/content/109/19/7241.abstract>.
- [31] W. Mason and D. J. Watts. "Financial incentives and the performance of crowds". In: *ACM SigKDD Explorations Newsletter* 11.2 (2010), pp. 100–108.
- [32] M. S. Bernstein, J. Brandt, R. C. Miller, and D. R. Karger. "Crowds in two seconds: Enabling realtime crowd-powered interfaces". In: *Proceedings of the 24th annual ACM symposium on User interface software and technology*. ACM. 2011, pp. 33–42.
- [33] M. J. Franklin, D. Kossmann, T. Kraska, S. Ramesh, and R. Xin. "CrowdDB: answering queries with crowdsourcing". In: *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*. ACM. 2011, pp. 61–72.
- [34] C. P. Ahn, R. Alexandroff, C. Allende Prieto, F. Anders, S. F. Anderson, T. Ander-ton, B. H. Andrews, É. Aubourg, S. Bailey, F. A. Bastien, et al. "The Tenth Data Release of the Sloan Digital Sky Survey: First Spectroscopic Data from the SDSS-III Apache Point Observatory Galactic Evolution Experiment". In: *The Astrophysical Journal Supplement Series* 211 (2014), p. 17.

- [35] C. J. Lintott, K. Schawinski, A. Slosar, K. Land, S. Bamford, D. Thomas, M. J. Rad-dick, R. C. Nichol, A. Szalay, D. Andreescu, et al. "Galaxy Zoo: morphologies de-rived from visual inspection of galaxies from the Sloan Digital Sky Survey". In: *Monthly Notices of the Royal Astronomical Society* 389.3 (2008), pp. 1179–1189.
- [36] P. B. Nair and R. G. Abraham. "A catalog of detailed visual morphological classi-fications for 14,034 galaxies in the sloan digital sky survey". In: *The Astrophysical Journal Supplement Series* 186.2 (2010), p. 427.
- [37] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. "Imagenet: A large-scale hierarchical image database". In: *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE. 2009, pp. 248–255.
- [38] C. Fellbaum. "WordNet: An electronic lexical database. 1998". In: *WordNet is avail-able from <http://www.cogsci.princeton.edu/wn>* (2010).
- [39] T. Volkmer, J. A. Thom, and S. M. Tahaghoghi. "Modeling human judgment of digital imagery for multimedia retrieval". In: *Multimedia, IEEE Transactions on* 9.5 (2007), pp. 967–974.
- [40] A. P. Dawid and A. M. Skene. "Maximum likelihood estimation of observer error-rates using the EM algorithm". In: *Applied Statistics* (1979), pp. 20–28.
- [41] J. Whitehill, T.-f. Wu, J. Bergsma, J. R. Movellan, and P. L. Ruvolo. "Whose vote should count more: Optimal integration of labels from labelers of unknown exper-tise". In: *Advances in neural information processing systems*. 2009, pp. 2035–2043.
- [42] P. Welinder, S. Branson, S. Belongie, P. Perona, and S. Diego. "The Multidimensional Wisdom of Crowds". In: (), pp. 1–9.
- [43] V. C. Raykar, S. Yu, L. H. Zhao, G. H. Valadez, C. Florin, L. Bogoni, and L. Moy. "Learning from crowds". In: *The Journal of Machine Learning Research* 99 (2010), pp. 1297–1322.
- [44] G. Demartini, D. E. Difallah, and P. Cudré-Mauroux. "ZenCrowd: leveraging prob-abilistic reasoning and crowdsourcing techniques for large-scale entity linking". In: *Proceedings of the 21st international conference on World Wide Web*. ACM. 2012, pp. 469–478.
- [45] A. P. Dempster, N. M. Laird, and D. B. Rubin. "Maximum likelihood from incom-plete data via the EM algorithm". In: *Journal of the Royal Statistical Society. Series B (Methodological)* (1977), pp. 1–38.
- [46] P. Smyth, U. Fayyad, M. Burl, P. Perona, and P. Baldi. "Inferring ground truth from subjective labelling of venus images". In: *Advances in neural information processing systems* (1995), pp. 1085–1092.

- [47] P. Donmez, J. G. Carbonell, and J. Schneider. "Efficiently learning the accuracy of labeling sources for selective sampling". In: *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 2009, pp. 259–268.
- [48] P. Welinder and P. Perona. "Online crowdsourcing: rating annotators and obtaining cost-effective labels". In: *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*. IEEE. 2010, pp. 25–32.
- [49] G. Chittaranjan, O. Aran, and D. Gatica-Perez. "Inferring truth from multiple annotators for social interaction analysis". In: *Workshop on Modeling Human Communication Dynamics at NIPS*. 2010, p. 10.
- [50] W. Tang and M. Lease. "Semi-supervised consensus labeling for crowdsourcing". In: *SIGIR 2011 Workshop on Crowdsourcing for Information Retrieval (CIR)*. 2011.
- [51] E. Kamar, S. Hacker, and E. Horvitz. "Combining human and machine intelligence in large-scale crowdsourcing". In: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*. International Foundation for Autonomous Agents and Multiagent Systems. 2012, pp. 467–474.
- [52] D. McDuff, S. Member, R. E. Kaliouby, and R. W. Picard. "Crowdsourcing Facial Responses to Online Videos". In: 3.4 (2012), pp. 456–468.
- [53] P. Ekman. "Facial expression and emotion". In: 48.4 (1993), 384–392.
- [54] Y.-L. Tian, T. Kanade, and J. Cohn. "Facial Expression Analysis". In: *Handbook of Face Recognition*. Springer New York, 2005, pp. 247–275. ISBN: 978-0-387-40595-7. DOI: 10.1007/0-387-27257-7_12. URL: http://dx.doi.org/10.1007/0-387-27257-7_12.
- [55] P. Ekman and W. Friesen. *Facial Action Coding System: A Technique for the Measurement of Facial Movement*. Palo Alto: Consulting Psychologists Press, 1978.
- [56] B. Borsboom. "Guess Who?: A game to crowdsource the labeling of affective facial expressions is comparable to expert ratings." In: ().
- [57] J. van der Schalk, S. T. Hawk, A. H. Fischer, and B. Doosje. "Moving faces, looking places: validation of the Amsterdam Dynamic Facial Expression Set (ADFES)." In: *Emotion* 11.4 (2011), p. 907.
- [58] Y.-Y. Chen, W. H. Hsu, and H.-Y. M. Liao. "Learning facial attributes by crowdsourcing in social media". In: *Proceedings of the 20th international conference companion on World wide web*. ACM. 2011, pp. 25–26.
- [59] Z. Zeng, M. Pantic, G. I. Roisman, and T. S. Huang. "A survey of affect recognition methods: Audio, visual, and spontaneous expressions". In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 31.1 (2009), pp. 39–58.

- [60] T. Kanade, J. F. Cohn, and Y. Tian. "Comprehensive database for facial expression analysis". In: *Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on*. IEEE. 2000, pp. 46–53.
- [61] N. Sebe, M. S. Lew, Y. Sun, I. Cohen, T. Gevers, and T. S. Huang. "Authentic facial expression analysis". In: *Image and Vision Computing* 25.12 (2007), pp. 1856–1863.
- [62] M. Pantic, M. Valstar, R. Rademaker, and L. Maat. "Web-based database for facial expression analysis". In: *Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on*. IEEE. 2005, 5–pp.
- [63] M. Pantic and M. S. Bartlett. "Machine analysis of facial expressions". In: (2007).
- [64] A. J. O'Toole, J. Harms, S. L. Snow, D. R. Hurst, M. R. Pappas, J. H. Ayyad, and H. Abdi. "A video database of moving faces and people". In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 27.5 (2005), pp. 812–816.
- [65] L. Yin, X. Wei, Y. Sun, J. Wang, and M. J. Rosato. "A 3D facial expression database for facial behavior research". In: *Automatic face and gesture recognition, 2006. FGR 2006. 7th international conference on*. IEEE. 2006, pp. 211–216.
- [66] H. Gunes and M. Piccardi. "A bimodal face and body gesture database for automatic analysis of human nonverbal affective behavior". In: *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*. Vol. 1. IEEE. 2006, pp. 1148–1153.
- [67] L. S.-H. Chen. "Joint processing of audio-visual information for the recognition of emotional expressions in human-computer interaction". PhD thesis. Citeseer, 2000.
- [68] G. I. Roisman, J. L. Tsai, and K.-H. S. Chiang. "The emotional integration of childhood experience: physiological, facial expressive, and self-reported emotional response during the adult attachment interview." In: *Developmental psychology* 40.5 (2004), p. 776.
- [69] M. S. Bartlett, G. Littlewort, M. Frank, C. Lainscsek, I. Fasel, and J. Movellan. "Recognizing facial expression: machine learning and application to spontaneous behavior". In: *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. Vol. 2. IEEE. 2005, pp. 568–573.
- [70] SAL. <http://emotion-research.net/toolbox/toolboxdatabase>. 2006-09-26.5667892524. 2005.
- [71] E. Douglas-Cowie, N. Campbell, R. Cowie, and P. Roach. "Emotional speech: Towards a new generation of databases". In: *Speech communication* 40.1 (2003), pp. 33–60.
- [72] M. Rokicki, S. Chelaru, S. Zerr, and S. Siersdorfer. "Competitive Game Designs for Improving the Cost Effectiveness of Crowdsourcing". In: *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*. ACM. 2014, pp. 1469–1478.