TÉCNICO LISBOA

Detecting Interaction Failures through Emotional Feedback and Robot Context

Fernando António Azinheira de Ramos Loureiro

Thesis to obtain the Master of Science Degree in

Engenharia Electrotécnica e de Computadores

Supervisors: Dr. Plinio Moreno Lopez
Prof. Alexandre José Malheiro Bernardino

Examination Committee

Chairperson: Prof. João Fernando Cardoso Silva Sequeira
Supervisor: Dr. Plinio Moreno Lopez
Member of the Committee: Dra. Joana Carvalho Filipe de Campos

Novembro 2021

Declaration

I declare that this document is an original work of my own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa.

Acknowledgments

I would like to acknowledge my dissertation supervisors Prof. Alexandre Bernardino, Prof. Plinio Moreno and João Avelino for their insight, support and sharing of knowledge that has made this Thesis possible.

To the E.A.S.S. and the family Santos Soares, whose without the support and scholarship I could not have gone this far in my studies.

Last but not least, to all my friends, colleagues and family that helped me grow as a person and were always there for me during the good and bad times in my life. Thank you.

To each and every one of you – Thank you.

Abstract

During human-robot interactions, robots may break social norms (Social Norm Violations - SNV) or perform erroneous behaviours due to sensor and actuator errors, and software issues (Technical Failures - TF). If robots are unaware of these errors, the interaction may become unpleasant or even risk user safety. While interacting, humans show various types of social signals that translate their inner state, which is concurrently estimated by other humans that detect social norm violations and react to them. To detect social errors and classify them as Social Norm Violations or Technical Failures, we propose to rely on Eye Gaze, Head Movement, Facial Expressions (Actions Units), and Emotions, as seen by the robot, along with the recent actions of the robot. We propose a two step cascaded decision, where the first step is to detect if an error occurs, followed by the error type classification (SNV vs. TF). We perform an extensive study of the various options on input data and classification algorithms, using a game-based scenario with a humanoid robot. We focus on Vizzy robot and in a dataset where Vizzy individually interacted with 24 participants in a block assembly game, where it had two moods. The "good" mood would help the participants win the game. The "bad" mood would be rude, causing social norm violations, and would clumsily destroy the assembled blocks, causing technical failures, and making the participant lose the game. Regarding the impact of input data, we observe that: (i) emotions improve the error detection step but not the error classification step, and (ii) the actions of the robot improves both error detection and error classification. Regarding the learning algorithms, Random Forest achieves the best performance both in error detection and error classification. The usage of the median filter on the error classification result increased the performance of Random Forest to 79.63% mean accuracy.

Keywords

Social Signals; Human-Robot Interaction; Emotions; Error Detection;Social Norm Violations; Technical

Failures.

Resumo

Durante as interações humanos-robôs, os robôs podem violar as normas sociais (Violações de Normas Sociais - SNV) ou realizar comportamentos errôneos devido a erros de sensor e atuador, e a falhas de software (Falhas Técnicas - TF). Se os robôs não estiverem cientes desses erros, a interação pode se tornar desagradável ou até mesmo colocar em risco a segurança do usuário. Enquanto interagem, os humanos mostram vários tipos de sinais que traduzem seu estado interno, que é simultaneamente interpretado por outros humanos que detetam violações das normas sociais e reagem a elas. Para detetar erros sociais e classificá-los como Violações de Normas Sociais ou Falhas Técnicas, propomos contar com Olhar, Movimento da Cabeça, Expressões Faciais (Unidades de Ações) e Emoções, vistas pelo robô, juntamente com as ações recentes do robô. Propomos uma decisão em cascata em duas etapas, onde a primeira etapa é detetar se ocorre um erro, seguida pela classificação do tipo de erro (SNV vs. TF). Realizamos um amplo estudo das várias opções de dados de entrada e algoritmos de classificação, usando um cenário baseado em jogo com um robô humanoide. Nós concentramos no robô Vizzy e em um conjunto de dados onde o Vizzy interagiu individualmente com 24 participantes em um jogo de montagem de blocos, onde havia dois humores. O "bom" humor ajudaria os participantes a vencer o jogo. O "mau" humor seria rude, causando violações das normas sociais, e destruiria desajeitadamente os blocos montados, causando falhas técnicas e fazendo com que o participante perdesse o jogo. Em relação ao impacto dos dados de entrada, observamos que: (i) as emoções melhoram a etapa de deteção de erros, mas não a etapa de classificação de erros, e (ii) as ações do robô melhoram a deteção de erros e a classificação de erros. Em relação aos algoritmos de aprendizagem, Random Forest obtém o melhor desempenho tanto na deteção quanto na classificação de erros. O uso do filtro de mediana no resultado da classificação do erro aumentou

o desempenho do Random Forest para 79,63 % de precisão média.

Palavras Chave

Sinais Sociais; Interação Humano-Robô; Emoções; Deteção de erros; Violação da Norma Social; Falhas Técnicas

Contents

List of Figures

List of Tables

List of Algorithms

Listings

Acronyms

HRI	Human Robot Interaction
HRC	Human Robot Cooperation
SNV	Social Norm Violation
TF	Technical Failure
EE	Execution Error
PE	Planing Error
AU	Action Units
SSP	Social Signal Processing
EEG	electroencephalogram
RF	Random Forest
NB	Naive Bayes
KNN	K-Nearest Neighbor

1

Introduction

Contents

1.1 Motivation

The idea that robots will be part of our daily life is becoming more and more realistic. Social robots will interact with us in many ways. For example, they will help us in education [3], in manufacturing [4] or even in assisting our elderly [5]. However, interaction failures can happen, either they are caused by a malfunction, Technical Failure (TF), or by a misunderstanding of the social conduct, that we use to guide our interactions, by the robot, Social Norm Violation (SNV) [6].

According to Salem et al. [7], people lost confidence in robots that showed unexpected actions in the context of the ongoing social interaction. Even though, some users continued to interact with the robot [8], which highlights security concerns. If the robot continues its harmful behaviour, it can endanger the user. For example, Morales et al. [9] conducted a study where the robot misbehaved, and even though the willingness to participate diminished, people still entered the workspace of the robot. These user behaviours point out the importance of the system to identify its own mistakes, to maintain user trustworthiness and safety. Thus, ensuring the correct behaviour and functionality of the robot, while interacting with humans, is fundamental.

A way to do this is by analysing the user feedback. Humans, while interacting with each other, use signals consciously and unconsciously that show their inner state [10]. For instance, a person can laugh, smile, or shake their head. Such signals vary according to the situation and the emotions felt. When interacting with robots, people also show these signals [11]. As such, a robot can use the capacity to detect user feedback automatically to verify and choose its actions. If a failure is detected, then the robot will be able to employ a recovery strategy to maintain the trust of the user at a certain level. Alternatively, the robot may implement a safety measure to avoid harming people around it.

1.2 Problem statement

In this work, we wish to answer the question: Can we identify the mistake of the robot by analysing the social signals of the user? In the context of a given social interaction, which social signals are more informative? There are many researchers that studied the latter question, where they analysed various responses from people that interacted with robots with erroneous behaviour (see more details at Chapter 2). These studies found the following promising social signals: eye gaze, head movement, facial expressions, speech, body movement, hand movement and bio-signals.

We focus on the subset of social signals which a mobile social robot can feasibly capture using its onboard sensors. Hence, we do not address bio-signals since they require invasive equipment. Moreover, we do not use speech features because automatic speech recognition (ASR) and natural language understanding (NLU) face challenges that affect their reliability on mobile robots. Nonetheless, most studies noticed that head movement, gaze shifting, and facial expressions/action units [12] [13] are the most

relevant social signals. As such, we focus on these signals. Furthermore, we also decided to use emotions, since, during human interaction, they also take an important part in communication, especially in an erratic situation [14]. For example, negative emotions, such as shame, guilt, or fear, tend to be very common when an error occurs in the workplace [15], or when teaching [16]. Additionally, we noticed that some studies mention the need of contextualizing the reactions of the users [17], [18]. As such, we decided to use the actions of the robot as the context, since there is a cause-effect relation between the actions of the robot and the reactions of the users.

Therefore, we now rephrase our question: Can we identify the mistakes of the robot by analysing the head movements, gaze patterns, facial expressions, emotions of the user and actions of the robot? Thus, our focus for this work is in creating an automatic algorithm that identifies error situations and classifies them as SNV or TF, based on the aforementioned social signals.

1.3 Challenges

Despite the general concordance of the most relevant social signals that people present during an error situation, they still differ in some ways from personality to personality [19]. The reaction of a person to an error is very susceptible to the context of the interaction, being difficult to interpret correctly due to ambiguities, especially smiling. For instance, Kontogiorgos et al. [18] used different groups of people for manual annotation of a dataset. There was an 85% agreement, showing that even humans cannot reach a consensus in understanding erroneous situations through behavioural signals.

Another issue to be considered is the lack of public datasets in human-robot interaction. Avelino and others [1] performed an experiment with Vizzy where it recorded a dataset with people reacting to its erratic behaviour. In our work, this is an important dataset since we will also be using Vizzy [20] for experiments and implementation of the algorithm.

1.4 Scope

Even though social robots are a rising theme, there still are not many people that have interacted with them. As such, it is important to notice that people tend to react in a very enthusiastic way or in a suspicious way when interacting with something they do not know or understand. In [21] the participants had not interacted before with a robot and therefore were more captivated with the technology and hence more patient with the mistakes. On the other hand, in [5] Vizzy interacted with elderly care residents, their first reaction was mostly freezing, but after getting familiarized with the robot, they started to become more active.

Since, in the general part of the studies, most experiments were performed with people that have not interacted with the robot before, we decided that our work will focus on error detection during zero-

acquaintance encounters between robots and people. Also, it is not yet very common to see social robots daily so it is plausible to assume that most of the interactions will be of this type.

Since people react differently when in the presence of a group, especially in terms of gaze shifting [22], [6], we decided that we will focus on interactions where only one person is interacting with the robot.

We focus on interaction with humanoid robots. During our work, we use the Vizzy robot [20].

1.5 Objectives

Our goal is to build an automatic algorithm to detect error situations during Human Robot Interaction (HRI), that works in real-time with signals captured by the onboard sensors of the robot. TWe study several types of input data to the algorithm. On the visual-based human perception, we consider: (i) Head pose, (ii) Gaze direction, (iii) Facial action units, (iv) Emotions. On the robot context information, we consider the following flags: (i) arm movement, (ii) speech. In addition, we consider previous action executed and its corresponding time. Regarding the classification algorithm, we study various options such as Random Forest, Isolation Forest, Naive Bayes, amongst others.

1.6 Outline

This document has a top-down approach, and the remainder is organized as follows: Section 2 - Related work, where we analyse various works whose subject is related to ours and help us in reaching a solution; Section 3 - Proposed Pipeline, where we present our solution, and talk about the means and tools to reach it; Section 4 - Error Detector Experiments; And Section 5 - Error Classifier Experiments, where we test the influence of each feature for our algorithm, and compare it to other solutions; Section 6 - Emotion Recognition, where we show in more detail our emotion recognition algorithm; And finally, Section 7 - Conclusion and Future Work.

2

Related work

Contents

Human Robot Interaction (HRI) is an area that has been in constant development. During interactions, robots should have mechanisms that ensure the trustworthiness and especially the safety of humans which is of the utmost importance. So, many researchers studied how people react towards an erroneous robot. These studies of human reactions are the key to define the principal features that should be taken into account in an algorithm so that the robot can identify these error situations.

2.1 Social Signals for Error detection

Giuliani et al. [6], analysed 201 videos collected from various projects. Most of the video corpus consists of interactions with mistakes, where the robot was assisting the participant with a task, such as building a wooden toy or serving drinks. According to this analysis, head movements and gaze shift were the most prevalent signals, especially looking back and forward between the robot and experimenter or a group member or the object. Participants frequently nod, shakes, or tilt their head. On the other hand, hand gestures and body movements were the least prevalent signals. Speech and facial expressions, especially smiling, were the second more relevant features. While trying to understand their results, they hypothesize that the reason for these signals can be due to the tasks performed by the participants. When interacting with the robot, participants "mostly stand still and do not show many body movements". Another found worth noticing is that people talked more during a social norm violation. In contrast, technical failures provoked more smiles. They also detected in many experiment videos that at the beginning of an error situation, the participants "froze", i.e. they kept standing still without moving. Finally, they highlight the importance of head movements in an automatic algorithm and the need to know if the experimenter or other humans are present since that influences the signals.

The same group of the previous paper, Mirning et al. [21], decided in this study to perform an experiment rather than analyse existent ones. Their experiment consisted of two phases. The first one was an interview, where the robot asked a few questions to the participant. The second was a more active task, where the robot asked the participants to build simple objects using LEGO. Throughout the experiments, they noticed that the type of error influences the type of social signal. Concerning the features, gaze shifting, facial expressions (laughter/smiling), and head movements were still more frequently made by people. However, this time body movement was also significant. They hypothesize that that is due to the activity of the task in the study being more active. In contrast to their previous study, people did not show many speech signals. Like before, they state that this is due to the nature of the task being more hands-on.

So far, the least probable social signals shown by people while in an erratic situation are body and hand movements. Even though the previous study analysed, presented, and argued that this is dependent on task, the most critical social signals remained the same throughout the two works above described. These are gaze shifting, head movement, facial expressions, especially laughing and smiling, and speech.

Another study that reached the same important features is [23]. Participants taught a robot how to dance, through learning from demonstration (LfD) setup and then observed the results. In error situations, participants usually shook their heads, frowned, lowered their heads with an adverted gaze, or closed their eyes for a long period of time. Participants especially showed a combination of smiling with head tilting and scrunched eyebrows when the robot made a mistake. While answering questions, some participants noted that towards the end, their patience had waned substantially, causing them to lose focus on the task. So, this study also shows that emotions of frustration and bore can be an indicator of something went wrong.

2.1.1 Non-humanoid robot

Until now, experiments shown were made using humanoid robots, which is also the case of our work. We now wonder, if the predominant social signals presented so far, can also be relevant when the social robot is non-humanoid. Stiber and colleagues [24] show that, despite using a robotic arm for their experiments, people did not react so differently in comparison to the previous papers. Speech was the most noticeable social signal, followed by gaze shifting, smile, head movement, laughter, scrunch, and brow raise. The experimenters believe that the presence of an overseer during the experiment was a contributing factor to the fact that people were talking and commenting so much during the error situations. They also noted that the sequence of social signals is important and that considering the order of social events might be of use. Finally, they state that reaction time and intensity may leverage the estimator of error severity.

2.1.2 Frequency and duration

The potential of the duration and frequency of the features has been pointed out by some studies.

Mirning et al. [25] continued their previous study [6], using the same dataset as before, but focusing more on how often people react, how long it takes, and which social signals are frequently shown together. Participants reacted faster to technical failures than to social norm violations. However, people show more social signals during the second type of error, which they also noticed in their previous study. During technical failures, surprisingly, people see less need to react. Also, concerning reaction times, some might have taken longer because people were "freezing", a social signal that was not taken into account in this study.

Cahya and colleagues [26] performed an experiment where they also focused on analysing the frequency and duration of the reactions of the participants. The experiment consisted of 3 phases, one interview, where the robot asked 5 questions, and 2 assembly sessions. In the first assembly session, the robot gave step-by-step instructions and in the second the person had to ask the robot for the missing parts of an assembly plan. There were in total of 50 participants. The error situations that they presented was mostly adopted from [21], with the difference that they divided technical failures into Execution

Error (EE), when a robot initiates the correct action but does it unsuccessfully, and Planing Error (PE), when a robot executes the action correctly, but it is the wrong action.

The experiment was recorded using RealSense D415 RGB-D camera and the social signals were annotated automatically using openFace [27]. With it, they managed to extract 17 types of facial action units, head orientation, head position, and gaze direction. The data shows that people show more facial expressions, head gestures, and gaze shifts during erroneous situations compared to error-free situations. More specifically, people tend to shift their gaze to the robot, and the study table more frequently. Also, they noticed that people tilt and move their heads forward more often during error situations.

In terms of duration, gaze shifts last longer in error situations.They hypothesize that the reason for eye gaze lasting so long is because people 'froze' during EE.

2.1.3 Smart Speakers

One of the constant features to be noticed in studies is speech, especially utterance and prosody. As such, studies, such as [28], that use smart speakers and search for error situations might be a good source of information. But for our problem, the work of Barkhuysen et al. [29] is a better example, since they combined speech and facial expressions to detect error situations during HRI with a smart-speaker. They noticed that people show head movements and frowning when aware of a communication problem. When they had to respond, hyper-articulation frequently occurs during an error situation. Smiling was also a sign of a problem, frequently accompanied by frowning or raising eye brown.

2.1.4 Bio-signals

During the search for the most current methods to identify error behaviour in the robot using human signals, we came across the usage of electroencephalogram (EEG) and other bio-signals [30], [31]. However, the technology to obtain these signals is quite invasive and implies that the user must be standing still and attached with wires, making it impractical to use. Nonetheless, it is an interesting area of this topic and one that perhaps, with better technology, could be used. For instance, train a robot with EEG and associate that signal with others easier to obtain, such as facial expressions.

2.2 Automatic algorithms for Error detection

To the best of our capabilities, we were only able to find 2 studies where they used an automatic method for detecting mistakes in the actions of the robot using the social signals of the participants as input features.

2.2.1 Random Forest

The main purpose of Kontogiorgos et al. [18], is to compare between smart-speaker and social robots and to investigate how robot embodiment affects the behavioural response to failures of the user.

They set up a Wizard-of-Oz[1] experiment where the participants were instructed to cook spring rolls by either the smart-speaker or a humanoid robot. For annotation, they used two in-lab annotators, where one watched the videos without audio, and the other with audio, and 192 online participants. The automatic detector consisted of a binary choice classifier (failure/not-failure). A Random Forest Classifier was implemented for that purpose. As features, they used gaze, head movements, and speech. The first two were obtained using a hat with motion-capture markers. For the latter they used the service Speech-To-Text of IBM Watson.

After the experiment, they noticed that head movement is the most important feature. Furthermore, speech tends to be the most relevant social signal when people interact with smart-speaker. And, when interacting with humanoid robots, gaze features are the most prominent social signals.

When comparing the various human annotators, Kontogiorgos et al. [18] also noticed that the one without audio performed the worst. Then they state that this highlights the importance of contextual information in assessing the response of people to robot failures. To finalize, for future work they recommend the addition of temporal information to the features.

2.2.2 NaiveBayes, Ibk and Rule learner Part

The main question of Trung et al. [19] is whether head and body movements, signals that can be tracked with RBG-D cameras, are enough to robustly detect errors.

This was the oldest paper in which we found an automatic algorithm, and they state that they did not find any related work with automated error detection from HRI researchers. Hence, we believe this one would be the first.

To collect the data between the interaction of the participants and a robot, that they programmed to make two types of errors (social and technical), they used the Kinect V1 RGB-D for skeleton and face tracking provided in Kinect SDK 1.8. The interaction consisted of two tasks, an interview, and a LEGO construction task. Then, they trained a rule learner **Part**, a **NaiveBayes** classifier, and **Ibk**, a k-nearest neighbour classifier, using six different sets of the collected data. These sets were meant to represent the movements of the participants and their temporal aspects. For evaluation they used 10-fold cross validation, to simulate a situation where the robot has seen the person before, and leave-one-out cross validation, to simulate a situation where the robot has never seen the person.

According to their results, body features (e.g. height) are not relevant to recognize an error, but the

[1] An experiment in which a subject is interacting with a computer system that is operated or partially operated by an unseen human

position the body takes is. The rule learner and k-nearest achieved a high classification rate ($> 90\%$) when having already data of the person. However, they performed poorly when not. Naive Bayes performed better in this situation.

They noticed that the social signals shown during error situations are different from person to person, and that could explain the better performance of 10-fold cross validation. Then they advise that the error detector should have a 2-stage process. First, identify the error and then verify if it's technical or social. For future work, they intend to use an outlier detector and add more modalities. For instance, check for smiles and laughter, and 'freeze'. Finally, after adding more features, a features selection algorithm, such as Correlation-Based Feature Selection, should be used to analyse the predictive capabilities of all features.

2.3 Emotion Recognition

Emotions are part of both human-human and human-robot interactions. The recognition of emotions is a topic that generates some interest in the HRI community and that has been thoroughly studied. In the studies mentioned in section 2.1, there were some that noticed how people changed their mood/emotions throughout the experiments with erroneous robots [26], [23]. As such, we decided to analyse and use emotion recognition for our error detection problem.

Most of the studies use the proposed idea of six basic emotions, happiness, anger, disgust, fear, sadness, and surprise [32]. The main feature in these studies is facial expression [33], [34], [35]. But there is also the increase of methods in emotion recognition using multimodal features such as body features [36], [37], and thermal features [34]. As we have seen before, the context can also be an important feature, [38] produced a publicly available automatic algorithm to identify emotions in an image considering the context shown, more specifically, the background of the image.

Facial expressions can also be translated into Action Units (AU), and some studies aim to associate specific action units to a corresponding emotion, more specifically the six basic ones [39], [40], [41].

2.4 Research Gaps

Even though there are numerous research on how people react during interaction with a faulty robot, only a few researchers applied those experiences to build an automatic algorithm to detect them. And, despite the numerous experiments, there are not many publicly available datasets with people interacting with a robot that makes mistakes.

We also noticed that emotions have not been used before to detect an error. However, Mirning and colleagues [21] did ask for the emotional state of the person during their experiments. And Cahya et al. [26] correlated their most relevant action units to their specific emotions (fear and surprise).

Thus, we intend to update the state-of-the-art with an automatic error detector algorithm that takes into consideration emotions and the context of the interaction.

3

Proposed Pipeline

Contents

3.1 Proposed Method

Our main goal is to build an algorithm to detect error situations and classify them as Social Norm Violation (SNV) or Technical Failure (TF). We propose the pipeline in Figure 3.1. Our algorithm detects and classifies errors frame by frame and, following Trung et. al. [19], does it in two steps. First, a Random Forest error detector uses robot context features, gaze features, head pose, facial action units, and emotions. If an error is detected, an error classifier, which is also a Random Forest model, uses all the previous signals except emotions to classify it as SNV or TF. Otherwise, the algorithm outputs the "No error" label. A median filter is also used on the error detector. This filter smooths the output by using past results to reject spurious miss-classifications. Our shortest reaction has a duration of around 2 seconds. As such, we decided that the window of the filter is about 30 frames, which is equivalent to one second.

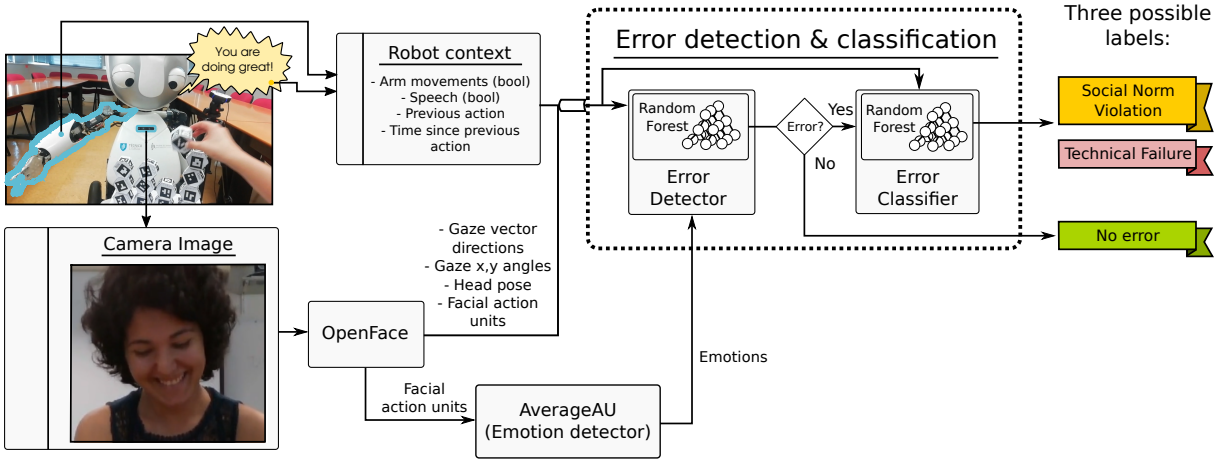


Figure 3.1: Proposed System

Based on the previous automatic error detector works [18] [19], we compare Random Forest with Naive Bayes and K-Nearest Neighbour in our problem. The reactions of the users to errors from the robot can be considered deviations from the normal behaviour during an interaction, as such, these reactions could be considered outliers. Thus, We also compare outlier detector methods with Random Forest.

In Algorithm 3.1 a pseudo-code that translates the general idea of the error detector is presented.

Algorithm 3.1: Error Detector

```
begin
    for each frame in video do
        [Gaze, Head, AU] = openFace(frame)
        Emotions = emotionAlgorithm(frame, AU)
        Action = getVizzyAction()
        if classifyError(Gaze, Head, AU, Emotions, Action) then
            classifySNVorTF(Gaze, Head, AU, Action)
```

3.2 Dataset

Our work addresses conditions where the robot (Vizzy) interacts with a single person (one-to-one interaction) during a first encounter. Thus, the robot has no prior information about the person. Also, we want to focus on data and experiments that happen in controlled environments, such as laboratories. For this work, we also focus on interactions where the participants follow instructions with the assistance of Vizzy, for instance during a block assembly game, during which Vizzy will have some functional failures (error situations SNV and TF).

Figure 3.2: Vizzy experiment from [1]

Vizzy [20] is a wheeled humanoid social robot with an anthropomorphic upper torso, that can navigate both indoors and outdoors. It is designed to interact with humans enjoyably, and combines easy mobility in planar surfaces, grasping ability, eye-head movements, and arm gestures.

With the lack of available datasets, we were planning in performing experiments with Vizzy, to collect more data. Unfortunately, due to the pandemic caused by covid-19, it is difficult to perform proper experiments to create a good quality dataset.

3.2.1 Vizzy dataset

We use the dataset of Avelino et al. [1], obtained in human-robot interaction experiments with the social robot Vizzy. The dataset consists of an experiment where 24 participants individually interacted with Vizzy in a block assembly game, where a video was captured from a camera on the robot.

Vizzy had two personalities/conditions, the "Kind robot" condition, and the "Grumpy robot" condition. It would help the participants win, if it was in the "Kind robot" condition, by giving them clues on where to find the missing parts. In the "Grumpy robot" condition it would make the participants lose the game, by destroying the construction with its arm, causing TF, and then blaming the participants, and tease them with sentences such as "you can get it right, but you annoy me by being so slow", generating SNV.

During the experiment with Vizzy, a laptop was also present to show the score and the rules of the

game. Because of this, during the interaction, the participants were not only interacting with Vizzy but also with the computer to verify the score, and start and finish of the game, Figure 3.3. The main goal of this work is to analyse data of people interacting with social robots, in our case Vizzy. As such, the experiments will focus on the usage of the Vizzy dataset angle alone, and the usage of the Vizzy dataset with the Laptop angle as an addition.

(a) **(b)**

Figure 3.3: Laptop Angle Sample

3.2.2 Dataset Annotation

Using openFace we automatically annotated each frame of the data to obtain the position and angle of the head, direction of eye gaze, and intensity and presence of action units. For each frame, openFace also outputted the confidence[1] of the landmarks it classified. Frames that had a confidence lower of 80% were discarded.

We annotated the actions of the robot and added the classes error, SNV and TF, which can be true or false, to each frame (the SNV and TF mistakes can happen simultaneously). We divided the actions of the robot into two categories, speak and move. We then labelled them as true on the frames where Vizzy moved or spoke (both actions can happen simultaneously).

To annotate the error and error type of the dataset we analysed the reactions of the users and labeled the beginning of an error as soon as we detected a reaction, and the end of the error when the reaction started to fade. Errors were classified as SNV or TF depending on the action of the robot. In the dataset of Avelino et al. [1], the majority of speak actions were SNV, while most move actions were TF. There were also some speak actions where the voice of Vizzy failed a bit and confused the participants, those were considered as TF. In Figure 3.4 we show the annotation of error and error type of a video from Avelino dataset, where we note that the dataset is imbalanced since the number of no error frames is significantly higher than the number of error frames (SNV and TF).

It is expected that this labelling method introduces some bias and error, because checking the exact starting frame and ending frame of a reaction is difficult. Moreover, since there was only one annotator

[1] https://github.com/TadasBaltrusaitis/OpenFace/wiki/Output-Format

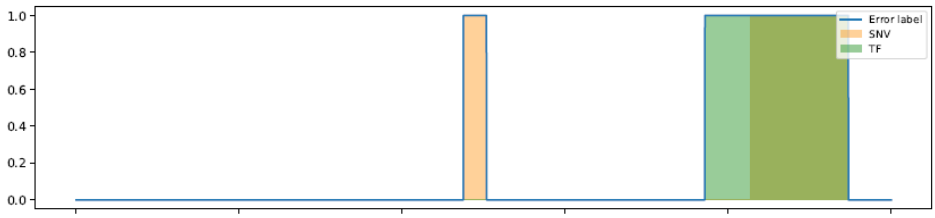


Figure 3.4: Error and error type annotation, blue line is the error (bool), orange shade is Social Norm Violation, green shade is Technical Failure

the length of the reaction, as well the reaction itself, can be biased.

3.3 Feature Extraction

To extract eye gaze, head movement and facial expressions(Action Units (AU)) we use openFace [27]. openFace is a state-of-the-art open-source facial behaviour analysis tool that is used by many researchers [26]. It is capable of facial landmark detection, head pose estimation, facial AU recognition, and eye-gaze estimation. The AU are values for individual components of muscle movements that brakes down facial expressions.

For emotion acquisition of the data, we propose a method that we called AverageAU. We compare it to some already built and trained emotion recognition algorithms. Surveys such as [42], and websites like *papers with code* [2] are a good source of information to know the current state of the art of emotion recognition tools, as well as to obtain the code made available of such works. The algorithms chosen to compare with are DeepFace [43], and Efficient CNN [44], which we describe in section 3.3.2. During the interaction between humans and robots, emotions should help in speeding up the process of detecting if something went wrong. We can assume that if the person is suddenly angry, scared, disgusted, sad, or surprised, then something happened that might be the fault of the robot [15].

3.3.1 Average of Action Units

To detect emotions, we propose a method that uses facial action units [45]. This way, we can use OpenFace to compute all head and face signals, with reduced computational requirements, since there is no need for an additional machine learning algorithm to obtain emotions.
By using the AU captured by openFace, we can associate them to the corresponding emotions [3] [40].
To do this, we performed the average of the action units for each of the emotions, equation 3.1. For instance, happiness is related to the AU12 and AU06, equation 3.2. Finally, we select the emotion with the highest average. A more detailed description and experiments of this algorithm are performed in

[2] https://paperswithcode.com/
[3] https://en.wikipedia.org/wiki/Facial_Action_Coding_System

Chapter 6, and in section 6.1 we show the various combinations of AU for each emotion.

$$Emotion = \frac{1}{n}\sum_{i=1}^{n} AU_i \tag{3.1}$$

$$Happy = \frac{AU12 + AU06}{2} \tag{3.2}$$

The neutral emotion is selected if the highest value is not above a previously defined threshold. We call this method AverageAU or avgAU.

3.3.2 DeepFace & Efficient CNN

DeepFace [43] is a lightweight face recognition and facial attribute analysis framework for python. It is a hybrid face recognition framework wrapping state-of-the-art models: VGG-Face, Google FaceNet, OpenFace, Facebook DeepFace, DeepID, ArcFace, and Dlib. It is capable to detect 7 emotions: Happiness, sadness, disgust, anger, fear, surprise, and neutral. DeepFace has become quite popular in the *github* community having over 130 thousand downloads. It is a new, constantly improving, and expanding algorithm. It achieved a 57% accuracy on the FER2013[4] dataset.

The work of Siqueira et al. [44][5], Efficient Facial Feature Learning with Wide Ensemble based CNN, is ranked as state of the art on papers with code for the FER+ dataset[6] achieving 80% accuracy. It outputs 8 emotions: Happiness, contempt, sadness, disgust, anger, fear, surprise, and neutral.

3.4 Actions of the Robot

Studies, such as [18], noticed that contextualizing the event could be important. In their case, they noticed that the annotators had issues in understanding an error situation when they did not have access to the sound of the video and could not listen to what the robot and the participants were saying (context). In our case, we define the context of the events as the action of the robot. The actions of the robot are what will, in the first place, provoke the reactions from the participants during the interaction. As such, we consider it the contextualization of the events.

The actions of the robot consist of the current action, which consists of movement (Boolean) and speech (Boolean), last performed action, which can be move, speech, or move&speech. And time since the last action, which is measured in seconds.

[4] https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge/data

[5] https://github.com/siqueira-hc/Efficient-Facial-Feature-Learning-with-Wide-Ensemble-based-Convolutional-Neural-Networks

[6] https://github.com/microsoft/FERPlus

3.5 Hypothesis Tests

To compare the proposed algorithm with other algorithms, we use the Wilcoxon test[7] [46], a non-parametric test that does not assume any properties regarding the distribution of the variables in analysis, and also the Student's t-test[8] [47]. The Wilcoxon test is used, over the t-test, when the distribution of the difference between the means of two samples cannot be assumed to be normally distributed. To test for normality, we used the Shapiro test[9] [48]. These hypothesis tests will tell us if there is a statistically significant difference between the algorithms: if the p-value is smaller than 0.05, then there is a statistically significant difference between the algorithms. Students t-test is said to be more reliable than Wilcoxon test when the assumption that the data has a normal distribution is assured [49]. On section 4.2.4 we compare both hypotheses and see that generally, both tests agree when a statistically significant difference is achieved.

The size effect is also used in some experiments when a statistically significant difference is achieved to evaluate the magnitude of the difference. We use the Cohen's d size effect [50]. If the d is below 0.2, then the size effect is small, if it is between 0.2 and 0.8 is considered medium, above 0.8 is considered large.

3.6 Proposed Pipeline Results

In this section, we show the results of our proposed solution for the error detector and classification, Figure 3.1. The error detector uses Random Forest with head, gaze, AU, emotions, and actions of the robot. The error classifier uses Random Forest with all the previous features except emotions. We compare it to the features used in previous automatic error detector works [18], [19], which used head and gaze features. First, we detect if an error has occurred with the error detector. If so, then the error type classifier is used. This is a multi-label problem with Error, SNV, and TF labels, hence, accuracy and hamming loss are used for the experiments.

In Figure 3.5 we show the results. The statistically significant difference is represented with: * == $p<0.05$, ** == $p<0.01$, *** == $p<0.001$, **** == $p<0.0001$. As we can see, our algorithm achieves a higher accuracy score of 72.77% while the head and gaze method achieved 57.21% with a statistically significant difference. Our method also achieves a lower hamming loss than the head and gaze algorithm. Cohen's d size effect was also used, achieving a large size effect of 5.24, meaning that the difference achieved has a large magnitude.

As such, our solution achieves better results in detecting and classifying an error than the one using only head and gaze features, features used in previous error detector works.

In Figure 3.6 we compare the proposed error detector and classifier with and without a median filter.

[7] https://en.wikipedia.org/wiki/Wilcoxon_signed-rank_test
[8] https://en.wikipedia.org/wiki/Student%27s_t-test
[9] https://en.wikipedia.org/wiki/Shapiro%E2%80%93Wilk_test

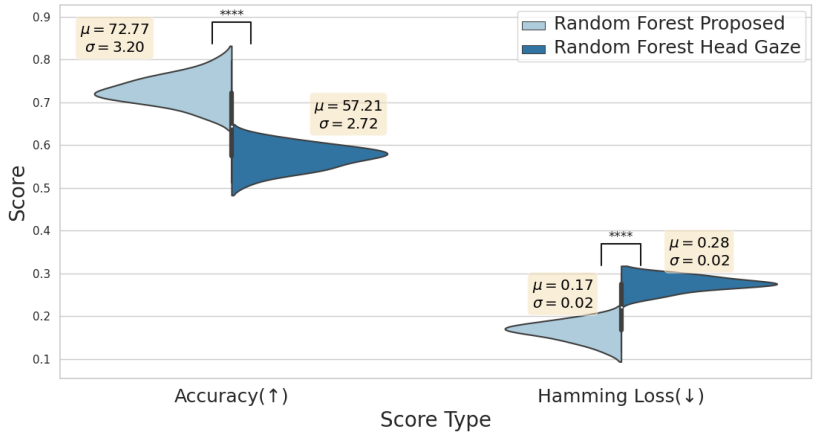


Figure 3.5: Comparing the proposed Error detector and classifier algorithm with the features used in previous works. ↑ - higher scores are better; ↓ - lower scores are better

As we can see, with the median filter the algorithm achieved a statistically significant different result of 79.63% mean accuracy, with a large size effect of 2.27.

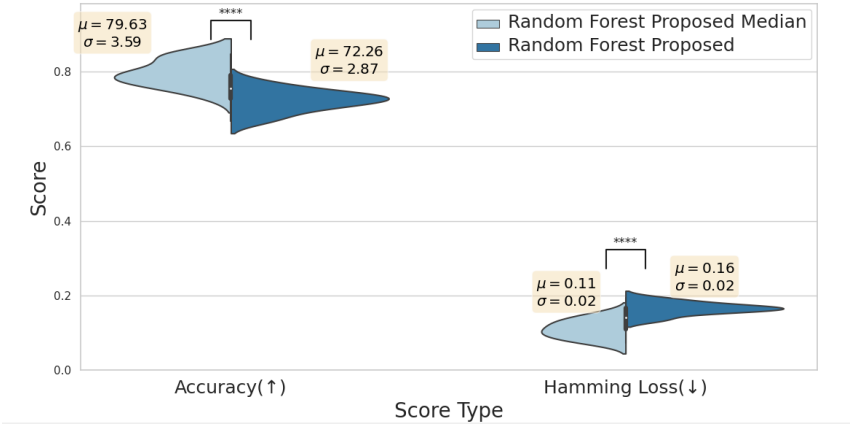


Figure 3.6: Comparing the proposed Error detector and classifier algorithm with and without median filter. ↑ - higher scores are better; ↓ - lower scores are better

In Figure 3.7 and Table 3.1 we show the results of the proposed error detector and classifier, using Vizzy and Laptop dataset. The algorithm achieved 67.69% mean accuracy and 0.20 mean hamming loss.

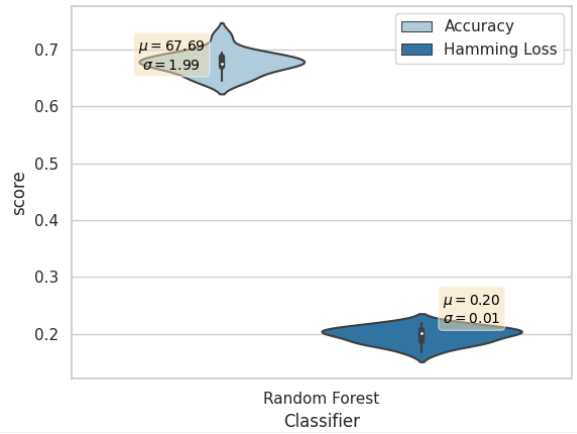


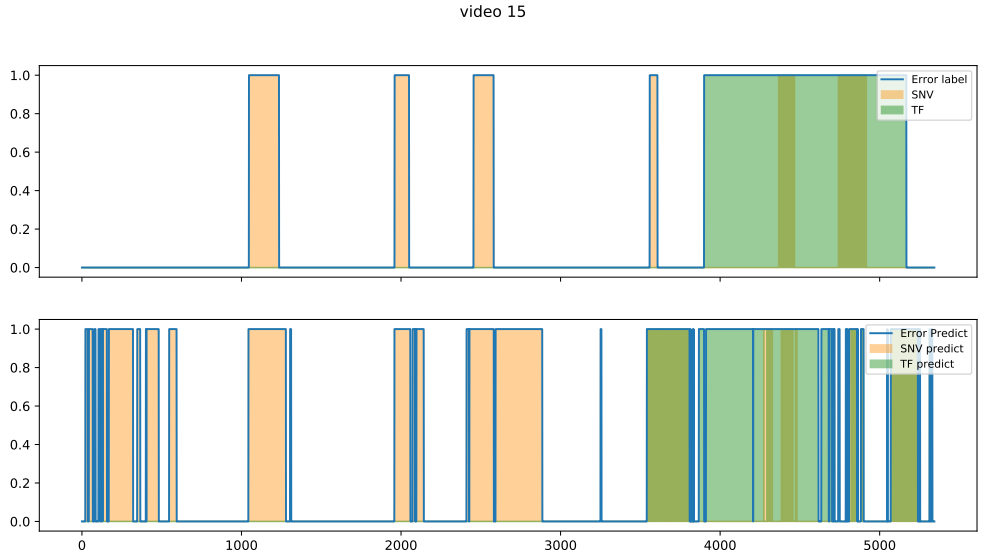
Figure 3.7: Multi-label accuracy and hamming loss of error detector and classifier combined, Vizzy and Laptop dataset

Individual Score		Accuracy	
		mean	SD
Error Detector		76.33	2.76
Error Classifier	SNV	78.62	2.75
	TF	73.74	3.53

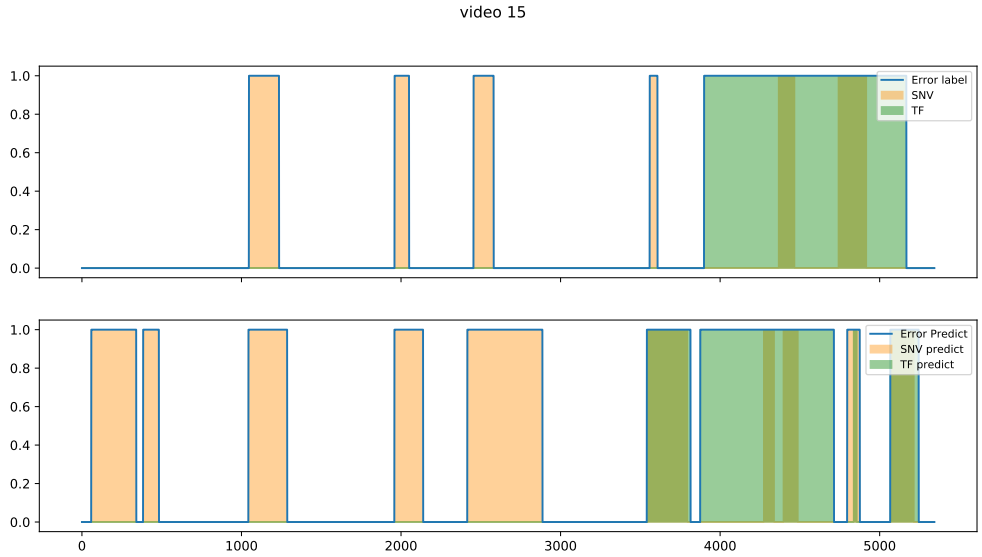
Table 3.1: Scores for each individual error detector and classifier, Vizzy and Laptop dataset

In the Figures 3.8 and 3.9, we show the detection of the error and its classification throughout interactions Vizzy had with one participant. We show the results with a median filter and without.

We can see that with the algorithm Vizzy can detect its own mistakes, even in cases that we did not notice (Figure 3.8, explained in more detail in section 4.1.6.A), and classify the mistake as Social Norm Violation and/or as Technical Failure.

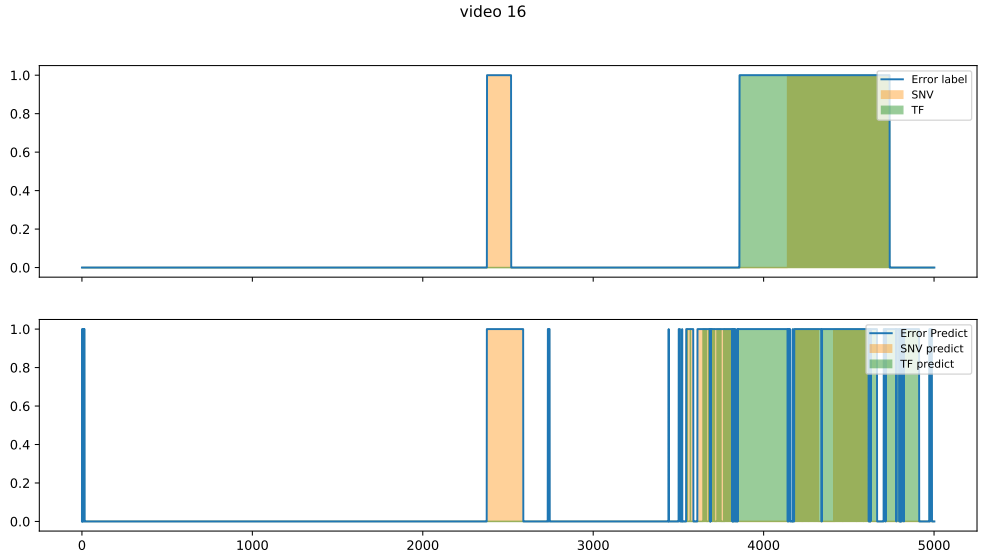


(a) No Median Filter

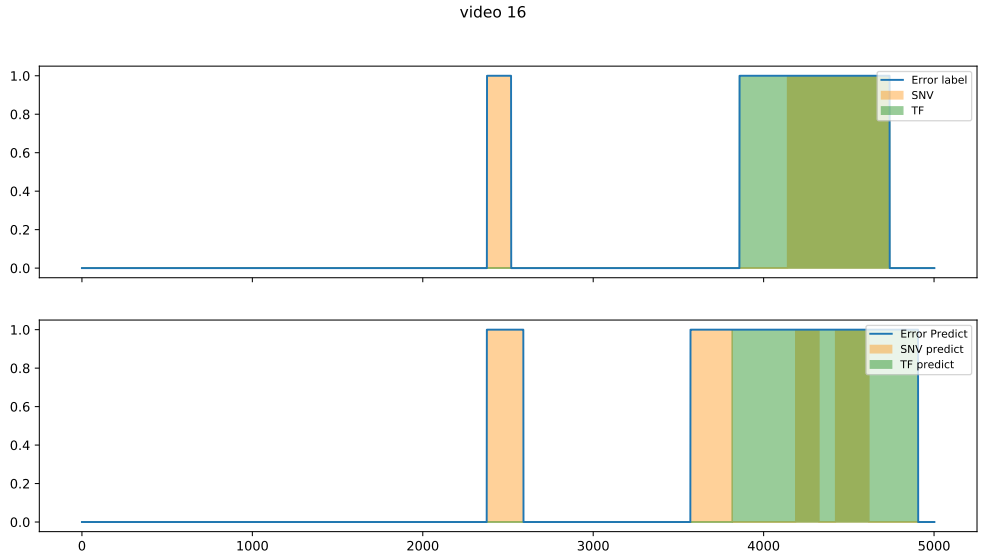


(b) Median Filter

Figure 3.8: Error detection (blue line) and classification of SNV (yellow) and TF (green) in an interaction. The upper graph is the Ground Truth.



(a) No Median Filter



(b) Median Filter

Figure 3.9: Error detection (blue line) and classification of SNV (yellow) and TF (green) in an interaction. The upper graph is the Ground Truth.

4

Error Detector Experiments

Contents

In this chapter, we describe and perform a set of experiments to evaluate the proposed error detector. To do so, we compare its performance against distinct combinations of classifiers and input features, where we use the accuracy and F1 score and check for statistically significant differences.

These experiments allow us to study the impact of the input features on the overall performance of the algorithm, allowing us to test the following hypotheses:

Hypothesis 1 (H1)**.** *Adding Facial Action Units to the literature base feature vector (head, gaze) will significantly improve error detection.*

Hypothesis 2 (H2)**.** *The addition of the current action of the robot to the literature base feature vector (head, gaze) will significantly improve error detection.*

Hypothesis 3 (H3)**.** *The addition of temporal information of the actions of the robot to the literature base feature vector (head, gaze) significantly improves error detection.*

Hypothesis 4 (H4)**.** *The addition of emotion information of the user to the literature base feature vector (head, gaze) significantly improves error detection.*

4.1 Error Detector

Based on the previous automatic error detector works [18] [19], we compare Random Forest (RF) with Naive Bayes (NB), and K-Nearest Neighbor (KNN). As planned, we start with the detection of the error, and then the classification of the error in SNV or TF.

An initial experiment was performed to understand which method to use for the balancing and splitting of the data. Regarding the balancing method, we tested under-sampling, and over-sampling. The under-sampling method consists of randomly choosing frames with errors classified as false. For the over-sampling, we performed data augmentation, namely horizontal flip, on the videos. However, even with this additional data, it was not enough to reach a balanced dataset, so under-sampling was still required, but now included more data. Nonetheless, we name this method over-sampling for comparison simplification.

For the splitting of the data, we used the train_test_split method from sklearn[1]. With this method distinct samples of the same video may appear during training and testing. We also used a random selector, where 25% of the videos of the dataset of Avelino et al. were for the test set. This way we are sure that the algorithm never sees people on the test set, during training.

In tables 4.1 and 4.2 we show the results of these experiments, where all the scores presented were obtained with an average of 10 runs for each condition. The base feature is the position and angle of the head, angle of gaze, and intensity and presence of AU.

[1] https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html

We begin by analysing the performance of the Random Forest when splitting the data with the test_train_split, Table 4.1. The algorithm achieved scores of around 98% accuracy. We hypothesize that these high scores point out that random forest can classify error situations correctly when it has seen the participant before.

Features	Balanced	Accuracy		F1		Average Precision	
		Mean	SD	Mean	SD	Mean	SD
Base	Under	98.11	0.14	98.11	0.14	97.10	0.27
Base	Over	98.28	0.091	98.28	0.091	97.31	0.17

Table 4.1: Results of Random Forest, using train_test_split

With the random selector, the error detector achieves lower scores than the other splitting method, Table 4.2. Both methods to balance the data achieved similar results, so we decided to use the data augmentation method for the following experiments.

Features	Balanced	Accuracy		F1		Average Precision	
		Mean	SD	Mean	SD	Mean	SD
Base	Over	74.74	5.96	73.64	6.80	71.94	5.67
Base	Under	73.71	3.88	72.58	4.77	70.74	3.25

Table 4.2: Results on Random Forest, randomly splitting the videos

We decided that for our experiments we will use the random selector for the splitting of the data since we focus on situations where the participants have not dealt with the robot before. As such, the robot has no previous data of that person. For the balancing of the data, we decided to use the data augmentation method.

4.1.1 Hyper Parameter Tuning

We tuned our Random Forest using grid search. The resulting best settings were n_estimators = 200, criterion= 'entropy', max_depth=10, max_features='sqrt', min_samples_leaf=15, min_samples_split=15. The default Random Forest achieved a 74.87% mean accuracy score, while the tuned version achieved a 77.81% mean score, with a statistically significant difference.
The section when we compare the different usage of features (section 4.2 to section 4.2.1) uses the default version of Random Forest.

4.1.2 Naive Bayes

In this section we compare Naive Bayes, a classifier that was used by [19], with Random Forest.

4.1.2.A Hyperparameter tuning

For our Naive Bayes we are going to use GaussianNB from scikit-learn[2]. With grid search, we found that the best settings for the GaussianNB are with a var_smoothing = 0.0028. The default Naive Bayes achieved a 67.76% mean accuracy score while the tuned version achieved a 77.94% mean accuracy score.

4.1.2.B Naive Bayes Vs Random Forest

In Table 4.3 and 4.4, we show the results of the experiment. We also present a comparison between the two more efficient combinations of features.

On Naive Bayes, the combination of the base plus the actions and emotions and base plus emotions achieved a Wilcoxon p-value above 0.05, and as such, they were not significantly different. On Random Forest, comparing the two combinations of features, the results are similar, and the p-value is above 0.05. Comparing Naive Bayes with Random Forest, Random Forest achieved better results, being statistically significantly different according to the Wilcoxon test.

Features	Classifier	Accuracy		F1		Average Precision	
		Mean	SD	Mean	SD	Mean	SD
+Actions +Emotions(1)	Naive Bayes	78.34	2.09	78.19	2.08	71.93	1.97
+Actions(2)	Naive Bayes	78.23	1.98	78.05	2.00	71.64	2.29
+ Actions +Emotions(3)	Random Forest	80.65	3.45	80.38	3.79	76.41	2.71
+Action(4)	Random Forest	80.66	3.58	80.39	3.92	76.44	2.82

Features	Wilcoxon Accuracy	
	p	stat
(1) Vs (2)	0.54	203.0
(1) Vs (3)	0.0003	57.0
(2) Vs (4)	0.0011	74.0
(3) Vs (4)	0.42	193.0

Table 4.3: Naive Bayes and Random Forest, on Vizzy angle dataset.

Focusing now on the usage of both the Vizzy angle and laptop angle, Table 4.4. Random Forest achieved higher accuracy mean scores, with a p-value below 0.05. When comparing the combination of features, in both algorithms, the p-value was always above 0.05, meaning that there was no significant difference between the two combinations.

Features	Classifier	Accuracy		F1		Average Precision	
		Mean	SD	Mean	SD	Mean	SD
+Actions +Emotions(1)	Naive Bayes	77.33	2.44	77.26	2.47	71.31	2.28
+Actions(2)	Naive Bayes	77.33	2.22	77.24	2.23	70.68	2.32
+ Actions +Emotions(3)	Random Forest	78.88	1.96	78.66	2.11	74.58	1.65
+Action(4)	Random Forest	78.77	1.91	78.55	2.05	74.44	1.71

Features	Wilcoxon Accuracy	
	p	stat
(1) Vs (2)	0.42	193.0
(1) Vs (3)	0.00049	63.0
(2) Vs (4)	0.0032	89.0
(3) Vs (4)	0.11	155.0

Table 4.4: Naive Bayes and Random Forest, on Vizzy and laptop angle dataset.

[2] https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.GaussianNB.html#sklearn.naive_bayes.GaussianNB

We can conclude that Random Forest achieves better results than Naive Bayes, as proposed. Between the usage of the base plus actions plus emotions and the usage of base plus actions, there was not a significant difference, when using Random Forest and Naive Bayes.

4.1.3 K Nearest Neighbour Vs Random Forest

Besides Naive Bayes, Trung et al. [19] also used a k-nearest neighbour classifier to detect error. In this section we compare Random Forest with KNN, both tuned.

In Table 4.5 we show the results. KNN proved to be computationally costly for our algorithm. Random Forest took around 10 seconds to fit and predict, while KNN took around 100 seconds. Nevertheless, according to the experiment, Random Forest achieves higher scores and is significantly different from KNN (p-value < 0.05).

Concerning the combination of features, on KNN there was no significant difference. However, on Random Forest the Wilcoxon test p-value achieved a value below 0.05, meaning that on these runs the usage of base plus actions and emotions performed the best since it reached a higher mean accuracy score with a lower variance.

Features	Classifier	Accuracy		F1		Average Precision	
		Mean	SD	Mean	SD	Mean	SD
+Actions +Emotions(1)	KNN	72.06	2.93	71.52	3.30	67.54	2.53
+Actions(2)	KNN	72.23	3.01	71.64	3.42	67.81	2.56
+Actions +Emotions(3)	Random Forest	79.91	3.14	79.66	3.36	75.97	2.70
+Actions(4)	Random Forest	79.56	3.22	79.29	3.43	75.59	2.88

Features	Wilcoxon Accuracy	
	p	stat
(1) Vs (2)	0.21	38.0
(1) Vs (3)	0.00065	0.0
(2) Vs (4)	0.00065	0.0
(3) Vs (4)	0.02	20.0

Table 4.5: KNN error detector, Vizzy angle

4.1.4 Random Forest Imbalanced data

We analyse how Random Forest behaves when classifying an imbalanced test set, with a balanced training set. To evaluate the results we used accuracy score, for the balanced test set, and the imbalanced data, we used Balanced Accuracy[3], a score also used by [18]. Besides these, we also look at F1-score, precision, sensitivity/recall, and specificity. We used only Vizzy dataset and show the results in Table 4.6. The balanced test set is obtained by sampling the imbalanced test set, meaning that both test sets have frames from the same video interactions.

Accuracy, balanced accuracy, sensitivity, and specificity reach similar results both in the balanced and in the imbalanced test set. F1 score and precision reach a lower score in the imbalanced test set. Concerning the hypothesis test to compare the two different features, in both test sets the same conclusion is reached, with a p-value above 0.05, that there was no significant difference between the two algorithms.

[3] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.balanced_accuracy_score.html

Features	Test Set	Accuracy/ Balanced Accuracy		F1 score		Precision		Recall/ Sensitivity		Specificity	
		Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
+Actions +Emotions(1)	Balanced	80.45	3.05	78.69	4.87	86.14	5.15	73.53	10.22	87.38	6.99
+Actions(2)	Balanced	80.41	3.23	78.63	5.26	85.90	4.87	73.66	10.68	87.16	6.67
+ Actions +Emotions(3)	Imbalanced	80.37	2.95	61.03	7.19	54.77	13.21	73.53	10.22	87.20	7.19
+Action(4)	Imbalanced	80.28	3.10	60.63	7.02	53.99	12.67	73.66	10.68	86.91	6.89

Features	Accuracy Hypothesis	F1 score Hypothesis
	p-value	p-value
(1) Vs (2)	0.80	0.80
(3) Vs (4)	0.60	0.25

Table 4.6: Comparison of results with balanced and imbalanced test set

4.1.5 Outlier Detection for Error Detector

While evaluating the distribution of emotions on error interactions and no error interactions we noticed that there is a different distribution of emotions, especially a spike of happiness, on the interactions with error, Figure 4.1. After evaluating these results and noticing that the response of the participants, when Vizzy made a mistake, is a deviation from the regular behaviour, we decided to try to use outlier detection algorithms to identify these mistakes.

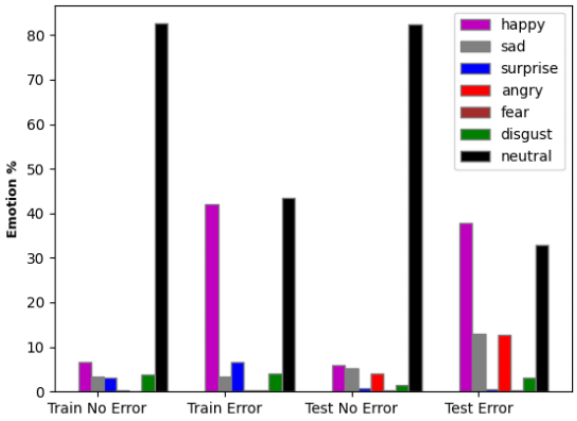


Figure 4.1: Distribution of emotions on train and test

We are interested in the following outlier detector methods, that are frequently used by the community [4] [5], and available on sklearn[6]:

- Isolation Forest [51]
- Local Outlier Factor (LOF) [52]
- One-class SVM [53]
- Minimum Convariance determinant (Elliptic Envelope) [54]

Domingues et al. [55] performed a comparison study on various outlier detection algorithms, where they concluded that Isolation Forest achieves better results in efficiently identifying outliers while showing

[4] https://machinelearningmastery.com/model-based-outlier-detection-and-removal-in-python/
[5] https://towardsdatascience.com/4-machine-learning-techniques-for-outlier-detection-in-python-21e9cfacb81d
[6] https://scikit-learn.org/stable/modules/outlier_detection.html#outlier-detection

excellent scalability on large datasets. One-class SVM also performed well, but it is not suitable for large datasets. Local Outlier Factor (LOF) reached the lowest performance.

4.1.5.A Training Set

The idea is that the frames that are labelled as errors are detected as outliers. Since we have those points labelled, we then have a case of supervised outlier detection. More specifically, we deal with novelty detection, which is similar to anomaly/outlier detection, but the models should be trained on a dataset free of anomalies [55]. Algorithms such as one-class SVM prefer the data to be as clean as possible, however, other unsupervised algorithms, such as Isolation Forest, can be fitted with a training set containing outliers, which in our case are what we labelled as error.
In this section, we compare the results when training Isolation Forest with data with no outliers, and with data with errors (outliers).

Training Set	Balanced Accuracy		F1 score	
	Mean	SD	Mean	SD
Clean(1)	74.98	2.23	50.69	6.45
Not Clean(2)	72.61	2.33	48.93	5.87

Features	Hypothesis
	p-value
(1) Vs (2)	0.0003

Table 4.7: Isolation Forest train with no error and with error

From Table 4.7, we can conclude that the outlier detector achieved higher scores when trained with a clean dataset, a data with no outliers/error. So, in the following experiments, we train the different outlier detectors with a clean training set.

4.1.5.B Tuning the outlier detectors

We tuned the outlier detectors and obtained the following parameters (parameters not mentioned are the default of sklearn):

- Isolation Forest: max_samples = 200, max_features = 15, n_estimators = 300, bootstrap = True, random_state = 16; (79%, t 6s)
- LOF: n_neighbors = 40, metric = 'euclidean'; (50%, t 300s)
- One-class SVM: Default; (55%, t 3000s)
- Elliptic Envelope: contamination = 0.2, random_state = 30; (75% t 200s)

4.1.5.C Comparing the outlier detectors

We now proceed to compare the outlier detectors. For this experiment, we use our dataset imbalanced. As such, we used the balanced accuracy and f1 score.
One-class SVM, as expected [55], does not behave well with large datasets, and is computationally costly,

as we have seen in the tuning section. Therefore, we will not be using this algorithm. On Table 4.8 we show the results. Isolation Forest was the outlier detector with the highest score.

Outlier Detector	Balanced Accuracy		F1 score	
	Mean	SD	Mean	SD
Isolation Forest(1)	**74.51**	2.64	**50.28**	4.54
Local Outlier Factor(2)	53.31	3.68	23.70	3.40
Elliptic Envelope(3)	69.43	5.65	41.54	7.05

Features	Wilcoxon
	p-value
(1) Vs (2)	5.96e-8
(1) Vs (3)	1.49e-6
(2) Vs (3)	5.96e-8

Table 4.8: Comparing different outlier detectors

4.1.6 Random Forest Classifier Vs Isolation Forest Outlier Detector

With the outlier detector chosen, in this section, we compare the Random Forest classifier and the Isolation Forest detector. For the comparison, we used imbalanced data on the test set, and make use of the student's t-test for the hypothesis test, when the data follows a normal distribution.
The experiments were done with the outlier detectors on the previous sections, were done using the base (head, gaze and AU) plus actions plus emotions features, in these experiments we will also see how the Isolation Forest behaves when using base plus actions.

In Table 4.9 we show the results of both algorithms when using Vizzy dataset. Starting with the different combinations of features on both algorithms the student's t-test, achieved a p-value above 0.05 meaning that the two combinations are not significantly different. Nonetheless, on both feature combinations, Random Forest classifier was the algorithm with the highest score, with (1) Vs (3) and (2) Vs (4) having a p-value below 0.05, on both scores, with a size effect above 0.8, meaning that Random Forest achieves a statistically significant different result from Isolation Forest, with a large size effect.

Features	Algorithm	Balanced Accuracy		F1 score	
		mean	SD	mean	SD
+Actions +Emotions (1)	Random Forest	80.13	3.22	62.68	1.78
+Actions(2)		79.96	3.49	62.15	1.72
+Actions +Emotions (3)	Isolation Forest	75.63	2.89	52.96	2.86
+Actions(4)		75.38	3.34	52.06	3.65

Features	Accuracy Hypothesis	Accuracy Size Effect	F1 score Hypothesis
	p-value	Cohen's d	p-values
(1) Vs (2)	0.14		0.15
(3) Vs (4)	0.12		0.06
(1) Vs (3)	1.73e-6	1.47	1.73e-6
(2) Vs (4)	2.35e-6	1.34	2.6e-4

Table 4.9: Random Forest Vs Isolation Forest, on Vizzy dataset

In Table 4.10 we show the results using Vizzy and Laptop dataset, where the conclusions are similar to the ones made above. Random Forest achieves better scores than Isolation Forest.

Features	Algorithm	Balanced Accuracy		F1 score	
		Mean	SD	Mean	SD
+Actions +Emotions(1)	Random Forest	76.88	2.28	61.34	4.48
+Actions(2)	Random Forest	77.01	2.31	61.49	4.31
+ Actions +Emotions(3)	Isolation Forest	69.44	2.00	48.84	4.12
+Action(4)	Isolation Forest	69.24	2.01	48.39	3.94

Features	Accuracy Hypothesis	F1 score Hypothesis
	p-value	p-value
(1) Vs (2)	0.8	0.72
(3) Vs (4)	0.36	0.08
(1) Vs (3)	6.10e-5	6.10e-5
(2) Vs (4)	6.10e-5	6.10e-5

Table 4.10: Random Forest Vs Isolation Forest, on Vizzy and Laptop dataset

With this, we can conclude, that for the error detector algorithm, Random Forest with Actions Units, Head position, and movement, Eye Gaze, Actions of the robot, and Emotions, is the better algorithm on our dataset, as proposed.

4.1.6.A Median

We intend in this section to verify how a median filter would improve both Random Forest and Isolation Forest. Since when an error happens, multiple error frames exist grouped, a single error frame or a small group of error frames is potentially a miss classified error. With this in mind, we hope to rectify these frames and improve our algorithm by applying a median filter to the output. The median filter has a window size of 30 frames.

Median	Algorithm	Balanced Accuracy		F1 score	
		Mean	SD	Mean	SD
No	Random Forest	79.53	3.27	61.65	3.92
Yes		80.30	3.59	63.08	3.80
No	Isolation Forest	74.34	2.65	51.90	5.89
Yes		75.35	2.86	53.77	6.42

Accuracy Hypothesis	F1 score Hypothesis
p-value	p-value
0.0002	6.10e-5
6.10e-5	6.10e-5

Table 4.11: Random Forest and Isolation Forest, with and without Median Filter, on Vizzy dataset

As we can see in Table 4.11, the addition of the median improves both algorithms, with a p-value below 0.05, meaning that there was a statistically significant difference between the two algorithms.

In figures 4.2 and 4.3, we show how both algorithms behave when detecting an error in one video recording of our dataset. With this, we hope to see in action the median filter, graphically.

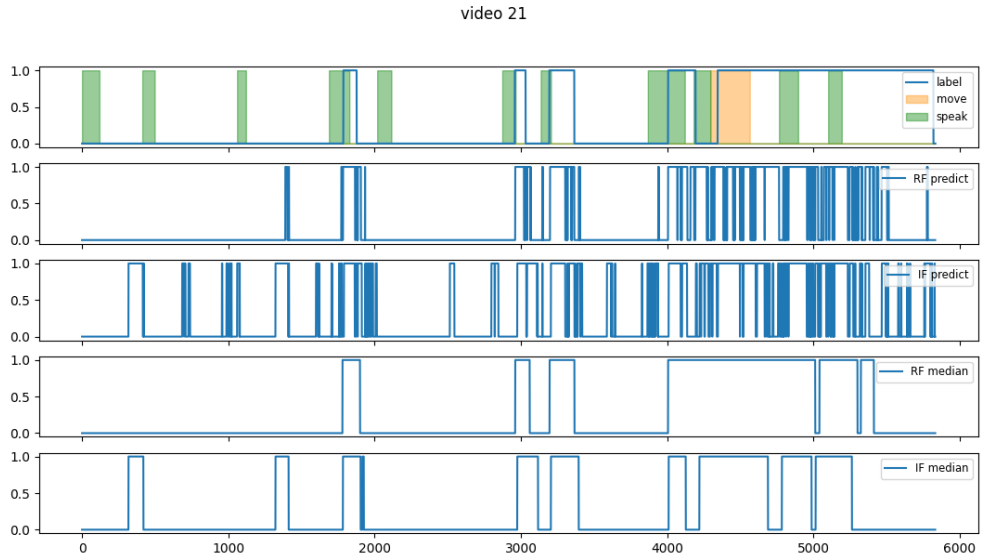


Figure 4.2: Error detector using Random Forest and Isolation Forest with and without Median Filter

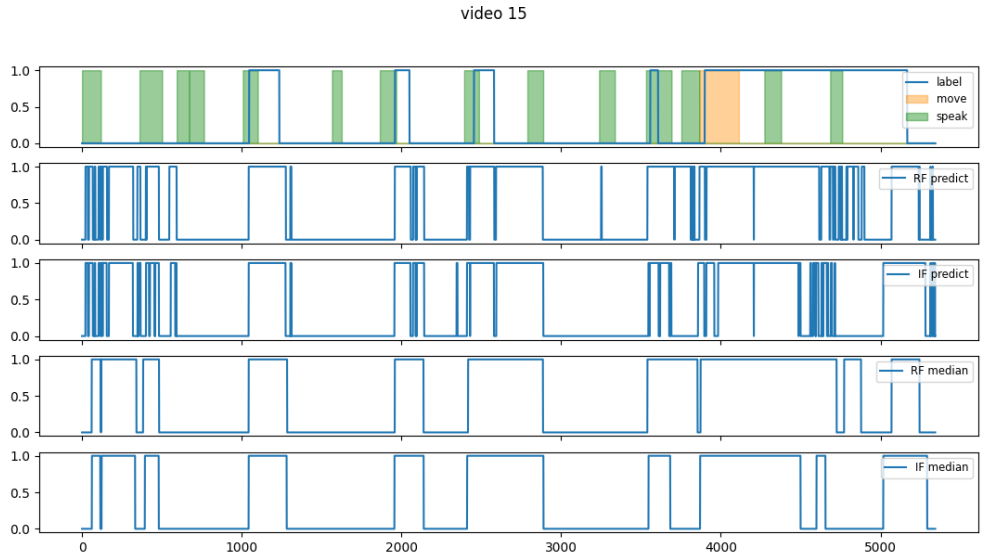


Figure 4.3: Error detector using Random Forest and Isolation Forest with and without Median Filter

In Figure 4.3 and Figure 4.2 the first plot line represents the labelled dataset, the blue line is the error, which is true when has value one, the green and yellow bars are when Vizzy speaks or moves, respectively. The second and third lines are the results from Random Forest and Isolation Forest, and the fourth and fifth lines are the results after the median filter. We can see that the median filter helps the algorithms on both figures.

However, at the beginning of the plot of Figure 4.3, there is no error labelled, meaning that we did not

find any reaction from this participant to the actions of the robot. It is important to remind that in this interaction most of the actions of the robot were intended to be mean and grumpy. It is then interesting that both algorithms detect an error situation in the same place after Vizzy spoke. This behaviour is noticeable in other videos as well. This event can be caused by noise in the data, or an actual reaction occurred which the algorithms detected, and could have been missed by us. Upon further inspection on the video in cause, the reaction detected by both algorithms was indeed an out of the ordinary one, but it was not a reaction to a mistake of the robot, but rather a reaction to a mistake that the user did and Vizzy corrected. On the other hand, when Vizzy corrected the mistake of the user, we did not consider it as rude, and hence did not label it as an error, but after evaluating it more, we can understand that for some people the straightforward correction of the robot could be considered as a social norm violation.

Even though Isolation Forest achieves lower scores over Random Forest, as an outlier detection method it has an advantage over Random Forest. Isolation Forest should be capable of detecting new types of reactions to mistakes, that Random Forest has not seen/trained yet. In the next chapter, we perform experiments to check this.

4.1.6.B Dealing with new mistakes

In this section, we test how Random Forest deals with new mistakes and compare them to Isolation Forest. For this experiment, we use violin plots to show the results. In Figure 4.4 we show the results with Random Forest trained with all the error examples from the dataset. As seen before, Random Forest achieved higher scores than Isolation Forest. Then, we remove from the training set the SNV errors, meaning that these types of mistakes are new on the test set. Figure 4.5(a) shows that in terms of accuracy there was no significant change, but when comparing F1 score, Random Forest still achieves higher scores with a statistical significance difference.
When removing the TF error type from the training set, Figure 4.5(b), Random Forest has a lower performance than Isolation Forest.

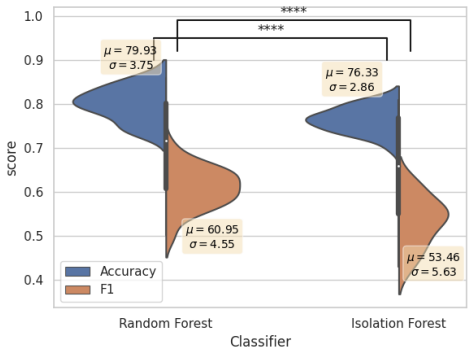
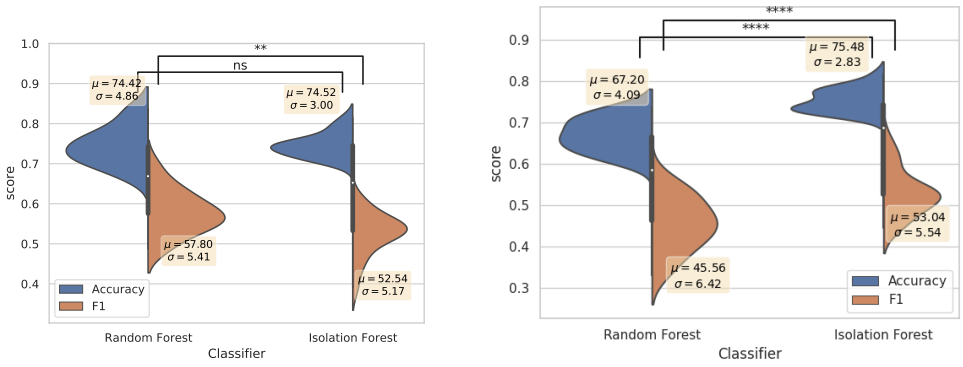


Figure 4.4: Comparing Random Forest with Isolation Forest, with the entire dataset



(a) Remove SNV errors

(b) Remove TF errors

Figure 4.5: Random Forest and Isolation Forest when dealing with new types of mistakes

This shows that the outlier detector method is more suitable when dealing with new mistakes. Regarding the removal of the SNV errors, Random Forest managed to perform better than Isolation Forest because it had the information of the reactions from the TF errors, which were more intense and distinctive reactions. This is also why Random Forest failed when we removed the TF errors, it had no relevant information to detect error.

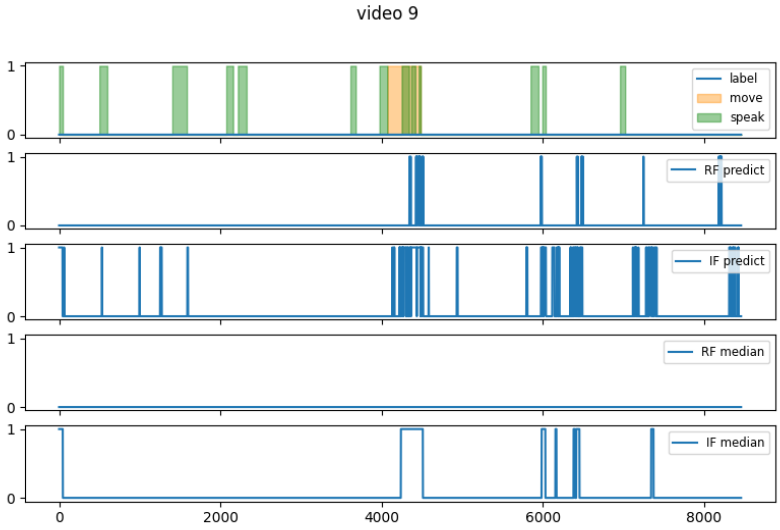


Figure 4.6: Error detector using Random Forest and Isolation Forest with and without Median Filter, On No Error Video

However, this advantage of the outlier detector when dealing with new error can also be a disadvantage. During the experiments we noticed that the Isolation Forest does not deal as well as Random Forest with interactions where no error occurs, resulting in more false positives. For instance, Figure 4.6 is an interaction with no error, where the outlier detector accused many situations as an error, and even

with the median filter it failed, while the Random Forest was more robust in this situation. This type of situation occurs frequently, on Fig 4.7 we show a violin plot where we can see that Random Forest achieves a statistically significantly different higher score than Isolation Forest when the test set only has no error situations.

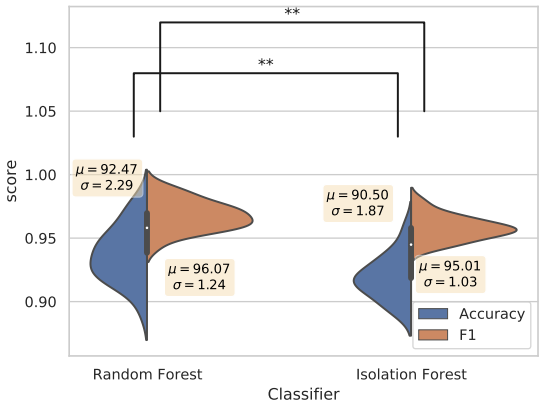


Figure 4.7: Comparing Random Forest with Isolation Forest, test set with only no error situations

4.2 Input features of the Error Detector

In this section, we compare how the different combination of features influences the algorithm. As mentioned before, the base is the use of Head, gaze, and AU features, to which we will then add the emotions, the actions of the robot, and then both.

In Table 4.12, we show the results of the various tests. We used accuracy, F1-score, and average precision score to evaluate the algorithms and run the hypothesis test for each score, to compare to the base algorithm. We can see that the addition of emotions did not cause a statistical difference, p-value above 0.05. But the addition of action and the addition of both actions and emotions, increased the mean of all scores, with a statistically significant difference ($p<0.05$). From this experiment, we can conclude then that the addition of actions of the robot helps Random Forest in detecting error situations, which confirms hypothesis **H2**.

Features	Accuracy		Wilcoxon		F1		Wilcoxon		Average Precision		Wilcoxon	
	Mean	SD	p	stat	Mean	SD	p	stat	Mean	SD	p	stat
Base	72.67	5.34			70.96	6.65			70.37	4.62		
+Emotion	72.62	5.34	0.7	214	70.90	6.66	0.7	214	70.32	4.54	0.64	210
+Action	72.94	5.28	0.017	166	71.27	6.57	0.018	168	70.63	4.47	0.015	162
+Action +Emotion	72.79	5.25	0.035	187	71.11	6.54	0.030	182	70.50	4.49	0.024	175

Table 4.12: Comparing different combination of features

With the use of both the data from Vizzy angle and the laptop angle, we proceed to another Wilcoxon test to compare the different combinations of features. All the additions achieved a p-value lower than 0.05, showing that they are significantly different from the base features. The addition of action and emotions to the base achieved the lowest p-value of the Wilcoxon test.

Features	Accuracy		Wilcoxon		F1		Wilcoxon		Average Precision		Wilcoxon	
	Mean	SD	p	stat	Mean	SD	p	stat	Mean	SD	p	stat
Base	70.01	3.02			67.95	4.01			67.86	2.59		
+ Actions	70.19	3.05	0.098	152	68.14	4.04	0.015	163	68.10	2.63	0.028	126
+ Emotions	70.56	2.88	0.003	90	68.63	3.78	0.0028	87	68.42	2.48	0.0041	93
+ Actions + Emotions	70.47	2.84	0.0023	84	68.50	3.73	0.0024	85	68.36	2.44	0.0020	82

Table 4.13: Comparing different combination of features, laptop and Vizzy view

It is worth noticing that even though we are mentioning the use of head, gaze, and AU as the base features, this is because these are the features that are considered the most relevant in the state of the art (chapter 2). However, the previous works that performed an automatic error detector ([19], [18]) only used head and gaze features. As such, we compare the improvement of the algorithm with the addition of the AU. We also check how the algorithm behaves with head, gaze, and emotions since the emotions are related to the AU. The results of these experiments are on Table 4.14

The addition of AU to the head and gaze features achieved higher scores, in comparison to the other two methods, and the Wilcoxon test resulted in a p-value below 0.05, showing that there is a statistically significant difference between the algorithms, proving the hypothesis **H1**. However, there is not a significant difference between the head and gaze features and the addition of emotions.

Features	Accuracy		Wilcoxon		F1		Wilcoxon		Average Precision		Wilcoxon	
	Mean	SD	p	stat	Mean	SD	p	stat	Mean	SD	p	stat
Head, Gaze (1)	58.96	3.64			53.80	6.01			56.83	3.00		
Head, Gaze, AU (2)	70.63	6.00			68.40	7.89			68.12	4.81		
Head, Gaze, Emotion (3)	58.21	5.27			54.26	6.97			54.30	5.33		
(1) Vs (2)			1.73e-6	0.0			1.92e-6	1.0			1.73e-6	0.0
(1) Vs (3)			0.48	198			0.20	170			0.15	170
(2) Vs (3)			1.73e-6	0.0			1.73e-6	0.0			1.73e-6	0.0

Table 4.14: Addition of action units

4.2.1 Temporal addition to the actions

The previous addition of actions in this section did not account for past information, using only data available on a single frame. As noted in past works [26], temporal information is relevant to detect whether an error has occurred. As such, we decided to add two features to represent some information about past events. The first feature encodes the last action (lastAction), with three possible values: speak, move,

and speak&move. The second feature represents how long the last action occurred (t_lastAction). The time starts to count as soon as the last action ends.

The experiments presented on tables 4.15 and 4.16, were both conducted using the random splitting of the videos, to ensure participants on the test set are not on the train set, with over-sampling.
In both tables, we compare the addition of actions and emotions to the base features, then the addition of the last action to the previous features, and finally the addition of the time since the last action to the previous features. The Wilcoxon test is performed by comparing the algorithm to the addition of action and emotions to the base, except when the column features say (2) Vs (3). In that case, the Wilcoxon test is performed with the addition of lastAction and the addition of lastAction and t_lastAction.
Comparing the addition of these temporal features, we can see that the algorithm is improved, and according to the Wilcoxon test the algorithms are significantly different ($p - value < 0.05$). When comparing the (2) Vs (3) we can also see that the p-values are below 0.05 with Vizzy angle, and when using both angles, indicating that these algorithms are also statistically significantly different.

Features	Accuracy		Wilcoxon		F1		Wilcoxon		Average Precision		Wilcoxon	
	Mean	SD	p	stat	Mean	SD	p	stat	Mean	SD	p	stat
+ Actions + Emotions(1)	73.43	6.61			71.84	8.04			70.80	5.79		
(1) + lastAction (2)	74.71	6.04	2.15e-5	21	73.38	7.24	2.16e-5	26	72.15	5.30	9.32e-6	17
(2) + t_lastAction (3)	75.85	5.80	1.24e-5	20	74.62	6.93	1.97e-5	25	73.62	5.08	2.35e-6	3.0
(2) Vs (3)			0.00066	67			0.0014	77			6.89e-5	39

Table 4.15: Addition of temporal actions, Vizzy view

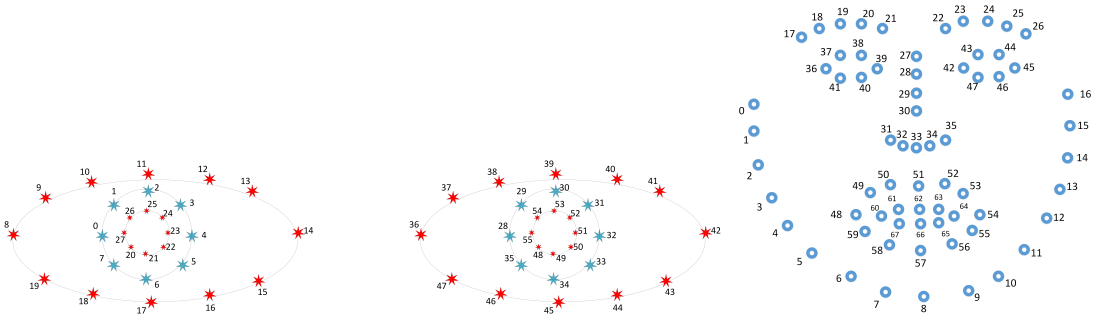
Features	Accuracy		Wilcoxon		F1		Wilcoxon		Average Precision		Wilcoxon	
	Mean	SD	p	stat	Mean	SD	p	stat	Mean	SD	p	stat
+ Actions + Emotions(1)	70.34	3.65			68.36	4.68			68.09	3.28		
(1) + lastAction (2)	71.98	3.69	1.73e-6	0.0	70.29	4.63	1.73e-6	0.0	69.82	3.34	1.73e-6	0.0
(2) + t_lastAction (3)	73.30	3.57	1.73e-6	0.0	71.76	4.41	1.73e-6	0.0	71.43	3.20	1.73e-6	0.0
(2) Vs (3)			2.88e-6	5.0			3.52e-6	7.0			1.73e-6	0.0

Table 4.16: Wilcoxon test with temporal actions, laptop and Vizzy view

From these experiments we can conclude that the addition of the temporal features improves the algorithm, verifying hypothesis **H3**.

4.2.2 Usage of more detailed features from openFace

Besides the features that we have been using from openFace: Head pose and orientation, eye gaze direction, and angle, and action units; OpenFace also outputs[7] landmark information regarding the eye region, face, and parameters of the rigid face and non-rigid face, Figure 4.8.



(a) Eye landmarks **(b)** Face landmarks

Figure 4.8: Landmarks

In this section, we added to our error detector the landmarks of the eyes and face. In Table 4.17 we present the results. The algorithm using all features (head, gaze, AU, emotions, actions) plus the landmarks achieved a mean accuracy of 69.34%, lower than the usage of the base plus actions and base plus actions and emotions, and is also significantly different (Wilcoxon p-value below 0.05). Besides not being as efficient as the other algorithms, it is also more computationally expensive since it increased the number of features from around 60 to 700.

Features	Accuracy		F1 score		Average Precision	
	Mean	SD	Mean	SD	Mean	SD
All + Landmarks (1)	69.34	3.88	67.88	4.82	65.96	3.51
+Actions(2)	75.78	3.16	75.20	3.66	72.08	2.76
+Action +Emotions(3)	75.96	3.23	75.40	3.72	72.22	2.82

Features	Wilcoxon	
	p	stat
(1) Vs (2)	1.73e-6	0.0
(1) Vs (3)	1.73e-6	0.0
(2) Vs (3)	0.086	138.0

Table 4.17: Random Forest error detector, with eye and facial landmarks

4.2.3 Error Detector for People with half mask

There are situations where the facial expression of a person is not visible or is partially hidden, for instance, with the current pandemic people use half masks to protect themselves and others. In this section, we performed experiments to compare how the error detector algorithm behaves when using

[7] https://github.com/TadasBaltrusaitis/OpenFace/wiki/Output-Format

features that can be obtained from these cases. As features, we used head pose and orientation, gaze direction and angle, and the actions of the robot, including the last performed action and time since that.

In Table 4.18 we show the results from the experiment, where the use of Gaze, Head, and Actions achieved a mean accuracy of 72.18%, a lower score when compared with the use of the other features, (2) and (3) from the table, but still an efficient algorithm.

Features	Accuracy		F1		Average Precision	
	Mean	SD	Mean	SD	Mean	SD
Gaze, Head, Actions (1)	72.18	4.25	71.39	4.73	68.22	4.09
Base +Actions(2)	79.86	3.06	79.57	3.32	76.00	2.58
Base + Actions +Emotions(3)	80.13	2.88	79.86	3.10	76.22	2.41

Features	Wilcoxon Accuracy	
	p	stat
(1) Vs (2)	1.73e-6	0.0
(1) Vs (3)	1.73e-6	0.0
(2) Vs (3)	0.026	124.0

Table 4.18: Error Detector with Features ready to deal with people with masks

4.2.4 Usage of Emotions on Error Detector

So far, besides comparing different classifiers we also have been comparing two combinations of features, base (head, eye gaze, and action units) plus actions and base plus actions plus emotions. It is noticeable from the experiments that the addition of the actions of the robot improves the algorithm in detecting error. However, when comparing the addition of actions to the addition of actions and emotions, there is not an obvious improvement. For instance, on Table 4.18 and 4.5 the p-value when comparing these combinations on Random Forest was below 0.05, meaning that on those runs they were significantly different, with the addition of emotions having a higher mean accuracy score. On the other hand, in the experiments shown in Table 4.4 and 4.3 the p-values of the Wilcoxon test were above 0.05, meaning that there was not a significant difference.

In this section, we decided to focus more on these two combinations of features and perform various runs to try to decide which one is better. Our dataset is composed of various video interactions of different participants interacting with Vizzy. On each run, we randomly select 25% of those videos to be the test set, and hence those participants do not enter the training set. So, what differs from run to run are the videos that are used to train both algorithms. Note that on the same run, both algorithms use the same training and test set. We performed 25 runs in total, with each run giving an accuracy score. We also used the McNemar test for each individual run, to test the output of both algorithms. This test will tell us that one algorithm makes more mistakes than the other, if the p-value of McNemar is below 0.05, and if it is above then the algorithms fail similarly.

We repeated this process 10 times, using only Vizzy view and then the combination of Vizzy and Laptop view.

The following tables show the results for each of the 25 runs. We show the mean accuracy score and standard deviation, the number of runs that the combination achieved better results, which was classified

as the one with the highest score when the McNemar test was below 0.05, and when this happened, we also recorded the maximum difference achieved, and the mean difference. This time we also use another hypothesis test, the Students t-test. This test is said to be more reliable than the Wilcoxon test when the assumption that the data has a normal distribution is assured, we will use both for comparison purposes. So, we will be using this test on the runs whose total accuracy scores output has a normal distribution. Finally, we performed the hypothesis test on all the mean accuracy scores of the 10 sets.

In Table 4.19 we show the results when using Vizzy angle. Looking at the Wilcoxon test and t-test we notice that there were 4 sets of 25 runs where the p-value achieved values below 0.05. On these sets, the highest mean accuracy score was from the combination of actions and emotions. On the hypothesis test that compared all the 10 sets of 25 runs, the p-value of both tests was below 0.05, meaning that the algorithms were significantly different, with the combination of the base plus actions plus emotions achieving a higher mean accuracy score.
So, we can conclude that using actions and emotions achieves better results. This verifies the hypothesis **H**4.

Set of 25 runs	Actions + Emotions (1) Score					Actions (2) score					Wilcoxon	t-test	hypothesis test for all 250 runs
	mean	SD	Runs	Max Diff	mean Diff	mean	SD	Runs	Max Diff	mean Diff	p-value	p-value	
1	79.99	3.21	13	1.30	0.73	79.77	3.41	3	2.37	1.50	0.11	0.30	
2	79.28	3.36	9	0.98	0.55	79.10	3.64	3	0.83	0.43	0.12	0.16	wilcoxon:
3	**81.00**	3.58	10	1.05	0.79	80.61	3.73	4	1.04	0.52	**0.0094**	**0.0091**	0.027
4	79.63	4.01	9	1.47	0.69	79.58	4.26	9	1.77	0.57	0.69	0.78	t-test:
5	**79.74**	3.09	9	2.05	0.79	79.27	3.35	4	0.47	0.22	**0.0027**	**0.0045**	0.016
6	**79.35**	3.33	11	1.24	0.62	79.11	3.45	4	1.30	0.64	**0.03**	**0.045**	(1):
7	80.15	3.11	13	1.56	0.75	80.06	3.48	9	2.87	0.89	0.38	0.66	79.99
8	**78.79**	3.19	17	1.76	0.67	78.44	3.51	3	1.90	1.32	**0.022**	**0.043**	(2):
9	81.95	3.19	8	0.66	0.49	82.18	3.40	8	1.93	1.05	0.34	0.17	79.80
10	79.99	3.07	16	1.17	0.56	79.89	3.41	7	1.62	0.95	0.31	0.54	

Table 4.19: Comparison between combination of features, with Vizzy dataset

Using now Vizzy and Laptop angle, Table 4.20, for each set there was one p-value below 0.05, and when checking the p-value of all the sets, it achieved values below 0.05, with the actions having a higher mean accuracy score on both cases.

Set of 25 runs	Actions + Emotions (1) Score					Actions (2) score					Wilcoxon	t-test	hypothesis test for all 250 runs
	mean	SD	Runs	Max Diff	mean Diff	mean	SD	Runs	Max Diff	mean Diff	p-value	p-value	
1	77.12	2.42	6	0.68	0.40	77.23	2.52	12	0.75	0.40	0.21	0.19	
2	76.46	2.58	6	0.39	0.24	76.54	2.65	9	0.74	0.25	0.47	0.86	wilcoxon:
3	76.65	3.30	8	0.60	0.34	76.73	3.45	9	1.19	0.62	0.71	0.44	0.007
4	78.60	3.47	10	0.76	0.22	78.72	3.47	10	1.23	0.50	0.23	0.18	t-test:
5	77.32	3.27	10	0.73	0.36	77.32	3.05	9	1.45	0.43	0.99	0.99	0.0003
6	78.18	3.94	7	0.86	0.50	78.24	3.97	14	0.90	0.31	0.38	0.50	(1):
7	76.32	3.81	8	0.89	0.26	76.45	3.90	10	1.61	0.50	0.22	0.21	77.09
8	76.16	3.36	3	0.66	0.33	**76.37**	3.54	16	1.40	0.41	**0.02**	**0.03**	(2):
9	77.43	2.92	7	1.59	0.65	77.53	2.94	12	0.92	0.55	0.22	0.44	77.19
10	76.65	3.23	10	0.84	0.22	76.81	3.35	8	0.90	0.43	0.46	0.16	

Table 4.20: Comparison between combination of features, with Vizzy and Laptop

With these experiments, we noticed that generally, both combinations are not significantly different. However, when using only Vizzy angle, the combination of base plus actions plus emotions features tends to achieve better results. When using both angles, the usage of base plus actions is the one that achieved slightly higher results.

We believe that with Vizzy view, using emotions helps the algorithm decide if there was an error or not because in the interactions where no error happens, most of the emotions are neutral. As such this helps the algorithm decides that there is no error if the emotion is neutral.

As for Vizzy with laptop angle, in the laptop angle, the participants are mostly in a profile angle, this makes it so that the action units from openFace are not as correctly calculated, which could explain the general lower accuracy scores when comparing when using only Vizzy angle, and the addition of emotions not helping the algorithm, since they are calculated with the action units.

In Conclusion, for the error detector the usage of Random Forest with head, gaze, AU, emotions, and actions of the robot achieves the best results, which confirms the hypothesis **H1**, **H2**, **H3**, and **H4**, as it was proposed in the pipeline.

5

Error Classifier Experiments

Contents

With the error detector built, we perform tests to identify the type of error that has occurred. The classification of the error as a TF or/and SNV is a multi-label classification problem because, at a certain time, both types of errors could have happened, and as such, an instance can be assigned with both.
For the score, since we are dealing with a multi-label classification, we are going to use the accuracy score and the hamming loss. The hamming loss outputs values between 0 and 1, the closest it is to zero, the better the algorithm.
As we did in the error detector, we start by comparing Random Forest with Naive Bayes and KNN. And then, we test different combination of features to analyze the proposed feature combination.

These experiments allow us to study the impact of the input features on the overall performance of the algorithm, allowing us to test the following hypotheses:

Hypothesis 5 (H5)**.** *Adding Facial Action Units to the literature base feature vector (head, gaze) will significantly improve error classification.*

Hypothesis 6 (H6)**.** *The addition of the current action of the robot to the literature base feature vector (head, gaze) will significantly improve error classification.*

Hypothesis 7 (H7)**.** *The addition of temporal information of the actions of the robot to the literature base feature vector (head, gaze) significantly improves error classification.*

Hypothesis 8 (H8)**.** *The addition of emotion information of the user to the literature base feature vector (head, gaze) significantly improves error classification.*

5.1 Error Classifier

We start by tuning the Random Forest classifier. We obtained the following hyperparameters: n_estimators=200, criterion='entropy', max_depth=40, max_features='sqrt', min_samples_leaf=1, min_samples_split=5. This led to an increase of 71% to 74% of the algorithm, with a Wilcoxon p-value below 0.05.

5.1.1 Naive Bayes Vs Random Forest

We start by comparing Naive Bayes, used by [19], with Random Forest.

The implementation of Random Forest on Scikit-learn can deal with multi-label problems, however, the Naive Bayes implementation can deal with multi-class but not with multi-label. As such, we need to use additional methods to test Naive Bayes. These methods can be OneVsRest multi-label strategy, Classifier Chain, or Label Powerset.
OneVsRest [1] uses binary relevance and assumes independence of the labels. Classifier Chain [56], assumes label correlation, and is an improved method of binary relevance.

[1] https://scikit-learn.org/stable/modules/generated/sklearn.multiclass.OneVsRestClassifier.html

In table 5.1 we show the results using the different methods with Naive Bayes. The classifier chain achieved the best result, as such it is the one that we are using with Naive Bayes for our multi-label classification problem.

Methods	Acuracy	
	mean	SD
OneVSRest	67.71	4.41
Classifier Chain	67.89	4.71
Label Powerset	64.12	1.01

Table 5.1: Multi label methods for Naive Bayes

We tuned the Naive Bayes and resulted with: var_smoothing=0.231; We then compared the tuned version and not tuned version and obtained the accuracy score of, 68% and 66%, respectively, with the Wilcoxon p-value test below 0.05.

In the following tables we show the results for the two classifiers, Naive Bayes and Random Forest, both tuned. For the experiment, we conducted 30 runs. Finally, we obtain the average and standard deviation of all 30 runs for the accuracy score and the hamming loss. We also display the accuracy of each label, SNV and TF.

Starting with Vizzy angle, table 5.2, Random Forest achieved the highest scores on accuracy mean score, and the lowest on hamming loss, with a p-value below 0.05, meaning that they were statistically significantly different. As such, Random Forest, when using only Vizzy angle, achieves better results on our dataset than Naive Bayes, as proposed.
Concerning the comparison of the two combinations of features, on both classifiers the Wilcoxon test showed that there was no significant difference when using base plus actions or base plus actions plus emotions.

Features	Classifier	Accuracy		Hamming Loss		SNV Accuracy		TF Accuracy	
		Mean	SD	Mean	SD	Mean	SD	Mean	SD
+ Actions (1)	Naive Bayes	70.80	8.82	0.178	0.0625	82.77	7.20	81.58	6.61
+Actions + Emotions(2)	Naive Bayes	70.98	7.71	0.180	0.0525	83.16	5.81	80.93	6.21
+Actions(3)	Random Forest	76.41	9.94	0.128	0.0551	85.97	4.82	87.74	9.47
+Action +Emotions(4)	Random Forest	76.11	10.26	0.130	0.0573	85.69	4.80	87.58	9.65

Features	Wilcoxon	
	p	stat
(1) Vs (2)	0.975	231.0
(1) Vs (3)	3.52e-6	7.0
(2) Vs (4)	3.72e-6	32.0
(3) Vs (4)	0.21	172.0

Table 5.2: Error type classifiers, Vizzy angle

Using Vizzy and Laptop angle, even though the accuracy is lower than before, the conclusions are the same. Random Forest was significantly different from Naive Bayes, with higher accuracy scores and lower hamming loss. And there was no significant difference in both classifiers when using either combination of features.

Features	Classifier	Accuracy		Hamming Loss		SNV Accuracy		TF Accuracy	
		Mean	SD	Mean	SD	Mean	SD	Mean	SD
+ Actions (1)	Naive Bayes	63.33	7.01	0.223	0.0378	83.22	2.61	72.11	6.08
+Actions + Emotions(2)	Naive Bayes	63.70	7.27	0.224	0.0382	82.11	3.16	73.01	5.75
+Actions(3)	Random Forest	66.67	8.11	0.182	0.0416	83.92	3.34	80.06	8.97
+Action +Emotions(4)	Random Forest	66.64	8.30	0.182	0.0427	83.70	3.32	80.16	9.09

Features	Wilcoxon	
	p	stat
(1) Vs (2)	0.21	172.0
(1) Vs (3)	0.0008	70.0
(2) Vs (4)	0.0057	98.0
(3) Vs (4)	0.99	232.0

Table 5.3: Error type classifiers, Vizzy and Laptop angle

We can then conclude that on our dataset Random Forest performs better than Naive Bayes, as proposed in the pipeline.

5.1.2 K Nearest Neighbour Vs Random Forest

In this section, we proceed to compare the usage of the K-nearest neighbour classifier(KNN), used by [19], and Random Forest. On the following tables we compare the usage of KNN with Random Forest, both tuned.

With Vizzy angle, table 5.4, Random Forest achieved higher accuracy mean scores and lower hamming loss, with a p-value below 0.05 when comparing with the KNN, meaning that the classifiers were significantly different, with Random Forest achieving better results.
About the two combinations of features, on KNN there was no significant difference, but on Random Forest the p-value was 0.006, with the usage of base plus actions achieving a higher result (72.46%) than base plus actions and emotions (71.83%).

Features	Classifier	Accuracy		Hamming Loss		SNV Accuracy		TF Accuracy	
		Mean	SD	Mean	SD	Mean	SD	Mean	SD
+ Actions (1)	KNN	64.53	7.32	0.220	0.044	74.31	5.13	81.65	6.43
+Actions + Emotions(2)	KNN	64.64	7.56	0.223	0.045	74.45	4.94	81.02	6.40
+Actions(3)	Random Forest	72.46	8.96	0.148	0.048	84.73	4.64	83.30	9.63
+Action +Emotions(4)	Random Forest	71.83	9.00	0.152	0.049	84.34	4.55	83.00	9.64

Features	Wilcoxon	
	p	stat
(1) Vs (2)	0.77	218.0
(1) Vs (3)	3.88e-6	8.0
(2) Vs (4)	3.52e-6	7.0
(3) Vs (4)	0.006	100.0

Table 5.4: Error type KNN Vs Random Forest, Vizzy angle

In table 5.5 we present the results using Vizzy and Laptop angle. As before, Random Forest is significantly different from KNN, with higher mean accuracy scores and lower hamming loss.
When comparing the two combinations of features, Random Forest was not significantly different, but KNN was, with base plus actions having a higher accuracy mean score and lower hamming loss.

Features	Classifier	Accuracy		Hamming Loss		SNV Accuracy		TF Accuracy	
		Mean	SD	Mean	SD	Mean	SD	Mean	SD
+ Actions (1)	KNN	59.34	6.47	0.257	0.043	73.46	4.86	75.16	4.96
+Actions + Emotions(2)	KNN	58.29	6.47	0.265	0.042	72.80	4.58	74.23	4.85
+Actions(3)	Random Forest	63.46	7.19	0.199	0.039	83.21	4.19	77.56	8.08
+Action +Emotions(4)	Random Forest	63.28	7.27	0.201	0.040	83.07	4.13	77.42	8.11

Features	Wilcoxon	
	p	stat
(1) Vs (2)	3.72e-5	32.0
(1) Vs (3)	2.88e-6	5.0
(2) Vs (4)	1.73e-6	0.0
(3) Vs (4)	0.5	200.0

Table 5.5: Error type KNN Vs Random Forest, Vizzy and laptop angle

We can then reach the conclusion that on our dataset Random Forest performs better than KNN, as proposed.

5.2 Input features of the Error Classifier

In this section, we perform experiments to compare the different combinations of input features, to evaluate the impact they have in the system in classifying error into SNV and TF.
Concerning the features, the base remains the head, gaze, and AU features, the +Emotion is the addition of the emotions, the +Action is the addition of the current action, last action, and time since last action.

On table 5.6 we show the results using Vizzy angle, and on table 5.7 we use Vizzy and laptop angle. When using only Vizzy angle dataset, all the additions achieved a p-value below 0.05 when compared to the base, meaning that the algorithms are statistically significantly different. Adding the actions achieved better results, however when comparing the addition of actions and the addition of actions and emotions, the p-value is above 0.05. As such, there is no statistically significant difference between these combinations of features. When using both angles, the addition of emotions to the base achieved a p-value above 0.05, so there is no statistically significant difference between these algorithms. The addition of actions achieved better results, but like before, there is no significant difference, according to the Wilcoxon test, between the addition of actions and the addition of actions and emotions.
With these results we can confirm the hypothesis **H6** and **H7**.

Features	Accuracy		Hamming Loss		SNV Accuracy		TF Accuracy	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Base (1)	62.23	9.13	0.244	0.068	70.42	7.45	80.78	9.72
+Actions(2)	74.56	8.47	0.136	0.045	86.38	4.08	86.46	8.88
+Emotions(3)	63.09	9.35	0.239	0.068	71.07	7.01	81.15	9.73
+Action +Emotions(4)	74.63	8.73	0.136	0.046	86.44	4.00	86.46	9.04

Features	Wilcoxon	
	p	stat
(1) Vs (2)	8.86e-5	0.0
(1) Vs (3)	0.033	48.0
(1) Vs (4)	8.86e-5	0.0
(2) Vs (3)	8.86e-5	0.0
(2) Vs (4)	0.68	94.0
(3) Vs (4)	8.86e-5	0.0

Table 5.6: Random Forest classify error type, combination of features, Vizzy angle

Features	Accuracy		Hamming Loss		SNV Accuracy		TF Accuracy	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Base (1)	55.29	8.57	0.283	0.058	66.94	6.14	76.49	7.16
+Actions(2)	66.88	6.99	0.183	0.038	83.38	3.91	80.11	5.78
+Emotions(3)	55.55	8.46	0.280	0.057	67.42	6.05	76.65	7.08
+Action +Emotions(4)	66.57	6.99	0.185	0.039	83.04	3.90	80.05	5.73

Features	Wilcoxon	
	p	stat
(1) Vs (2)	8.86e-5	0.0
(1) Vs (3)	0.295	69.0
(1) Vs (4)	8.86e-5	0.0
(2) Vs (3)	8.86e-5	0.0
(2) Vs (4)	0.14	65.0
(3) Vs (4)	8.86e-5	0.0

Table 5.7: Random Forest classify error type, combination of features, Vizzy and laptop angle

We proceed to evaluate the addition of the AU to the gaze and head, features used in previous studies. According to the Wilcoxon test, when combining Vizzy and laptop angle, there was not a statistically significant difference with the addition of the action units, table 5.8. As we mentioned before, we hypothesize that a reason for the results obtained when using the laptop view being lower is that in that point of view the participants are looking sideways, so openFace has a higher difficulty in correctly acquiring the action units of the participants, this experiment is also an indication of this.
When only using Vizzy angle, there is a statistically significant difference between the algorithms, with the addition of the action units improving the algorithm, table 5.9. This confirms the hypothesis **H5**.

Features	Accuracy		Hamming Loss	
	Mean	SD	Mean	SD
Head and Gaze (1)	58.33	5.53	0.283	0.042
Head, Gaze and AU(2)	59.31	6.45	0.268	0.043

Features	Wilcoxon	
	p	stat
(1) Vs (2)	0.079	58.0

Table 5.8: Compare the addition of Action Units, Vizzy and laptop angle

Features	Accuracy		Hamming Loss	
	Mean	SD	Mean	SD
Head and Gaze (1)	59.90	9.80	0.27	0.073
Head, Gaze and AU(2)	64.11	10.0	0.24	0.073

Features	Wilcoxon	
	p	stat
(1) Vs (2)	0.0001	5.0

Table 5.9: Compare the addition of Action Units, Vizzy angle

In the next experiment, we compare the use of only the actions with the previous combinations, table 5.10. The results indicate that to classify the type of error when using both Vizzy angle and laptop angle, only using actions achieves a lower accuracy score. However, there is no significant difference between the algorithms. With only Vizzy angle, using the combination of features is better than only using actions, table 5.11.

Features	Accuracy		Hamming Loss	
	Mean	SD	Mean	SD
Only Actions (0)	64.64	2.63	0.222	0.014
Base + Actions(1)	67.42	5.27	0.179	0.040
Base + Actions + Emotions (2)	67.61	5.36	0.181	0.041

Features	Wilcoxon	
	p	stat
(0) Vs (1)	0.10	61.0
(0) Vs (2)	0.062	55.0
(2) Vs (1)	0.16	67.0

Table 5.10: Experiments with the features of Action, Vizzy and laptop angle

Features	Accuracy		Hamming Loss	
	Mean	SD	Mean	SD
Only Actions (0)	70.08	2.72	0.20	0.0153
Base + Actions(1)	76.67	8.65	0.127	0.048
Base + Actions + Emotions (2)	76.44	8.59	0.128	0.048

Features	Wilcoxon	
	p	stat
(0) Vs (1)	0.009	35.0
(0) Vs (2)	0.009	35.0
(2) Vs (1)	0.35	72.0

Table 5.11: Experiments with the features of Action, Vizzy angle

5.2.1 Usage of Emotion for Error Classifier

Throughout the error classifier experiments, there was not a clear observation if the addition of emotion to the actions improved the algorithm. So, in this section, we present an experiment similar to the one done in the error detector, where we perform 10 set experiments, each constituted by 25 runs. To compare the algorithms, we used the Wilcoxon test and the student's t-test. To use the t-test the data has to follow a normal distribution, so we only used the results where this happened. To verify if the score were normal distribution, we performed the Shapiro test.

We used only Vizzy angle first, table 5.12, and then both angles Vizzy and Laptop, table 5.13. In both cases the conclusions are similar. the usage of base plus actions more frequently achieves the highest accuracy scores with a statistically significant difference. The overall hypothesis test also achieved a p-value below 0.05, with the best performance achieved by the base plus actions of 75.48% accuracy. This negates the hypothesis **H8**.

Set of 25 runs	Actions + Emotions (1) Score		Actions (2) score		Wilcoxon	t-test	hypothesis test for all 250 runs
	Accuracy	Hamming	Accuracy	Hamming			
	mean (SD)	mean (SD)	mean (SD)	mean (SD)	p-value	p-value	
1	74.58 (7.98)	0.133 (0.041)	**75.43 (7.90)**	0.128 (0.040)	**0.00055**	**0.00058**	
2	73.79 (8.62)	0.14 (0.053)	**74.20 (8.54)**	0.137 (0.052)	**0.027**	**0.016**	wilcoxon:
3	75.52 (7.60)	0.13 (0.04)	**76.15 (7.58)**	0.127 (0.04)	**0.01**	**0.008**	0.0019
4	74.12 (10.99)	0.14 (0.06)	74.47 (10.96)	0.139 (0.062)	0.28	0.22	t-test:
5	72.18 (9.55)	0.149 (0.053)	**72.93 (9.69)**	0.144 (0.052)	**0.0009**	**0.0002**	0.014
6	76.05 (8.28)	0.129 (0.045)	**76.86 (7.96)**	0.124 (0.043)	**0.0002**	**0.0003**	(1):
7	73.28 (8.33)	0.142 (0.044)	73.51 (9.07)	0.14 (0.047)	0.24	0.44	74.65
8	77.02 (9.11)	0.123 (0.047)	**77.80 (9.19)**	0.118 (0.047)	**0.0037**	**0.0079**	(2):
9	75.82 (6.93)	0.113 (0.035)	**79.03 (6.94)**	0.11 (0.034)	**0.011**	**0.009**	75.48
10	74.20 (9.04)	0.139 (0.05)	74.46 (9.43)	0.136 (0.05)	0.29	0.30	

Table 5.12: base + actions Vs base + actions + emotions on Vizzy angle

Set of 25 runs	Actions + Emotions (1) Score		Actions (2) score		Wilcoxon	t-test	hypothesis test for all 250 runs
	Accuracy	Hamming	Accuracy	Hamming			
	mean (SD)	mean (SD)	mean (SD)	mean (SD)	p-value	p-value	
1	64.67 (7.15)	0.188 (0.037)	64.95 (7.41)	0.186 (0.038)	0.14	0.16	
2	66.74 (5.64)	0.179 (0.03)	67.10 (5.30)	0.176 (0.028)	0.07	0.1	wilcoxon:
3	67.09 (6.08)	0.176 (0.032)	67.30 (6.39)	0.176 (0.035)	0.19	0.13	0.0019
4	69.12 (5.98)	0.167 (0.032)	**69.46 (6.08)**	0.165 (0.032)	0.06	**0.036**	t-test:
5	66.45 (7.16)	0.179 (0.038)	66.73 (7.06)	0.178 (0.038)	0.33	0.21	1.66e-6
6	70.59 (5.59)	0.159 (2.85)	**71.00 (5.71)**	0.156 (0.029)	**0.03**	**0.03**	(1):
7	68.55 (7.02)	0.169 (0.037)	**69.12 (6.90)**	0.166 (0.036)	**0.003**	**0.01**	67.12
8	64.86 (6.62)	0.187 (0.035)	65.21 (6.84)	0.185 (0.036)	0.12	0.11	(2):
9	65.79 (8.12)	0.181 (0.04)	**66.27 (8.15)**	0.178 (0.04)	**0.02**	**0.04**	67.48
10	67.31 (7.63)	0.174 (0.039)	67.64 (7.65)	0.173 (0.04)	0.08	0.14	

Table 5.13: base + actions Vs base + actions + emotions on Vizzy and Laptop angle

Lastly, the usage of emotions helps in detecting an error but not in classifying its type. We hypothesize that that is due to the fact that, in the dataset used, there is a distinct change of emotional state of the participants when an error occurs. However, the type of emotional reaction is similar for both types of errors. For instance, people look happy or surprised in both SNV or TF. As such, the emotions do not help in classifying the type of error.

We conclude then that for the error type classifier we use Random Forest with head, eye gaze, actions units, current action, last action and time since last action as features, which confirms the hypothesis **H5**, **H6**, and **H7**.

6

Emotion Recognition

Contents

In this chapter, we perform a set of experiments to evaluate the proposed emotion detector, averageAU, and validate its use over alternative methods. Due to the widespread use of facial half masks due to the COVID-19 pandemic, we also test the emotion detector algorithm in masked faces.

To evaluate the proposed emotion recognition method, averageAU, we perform three experiments. First, we compare the combination of AU proposed by Ekman et al. [45] with other combinations proposed by Ghayoumi et al. [40], Lucey et al. [57] and Karthick et al. [58]. We also test several thresholds for neutral emotion. In the second experiment, we compare AverageAU with DeepFace [43] and Efficient CNN [44]. In the third, we test the applicability of our algorithm when dealing with people using half-face masks.

The AverageAU is a proposed algorithm where each emotion has a value that is obtained by the average of their specific corresponding actions units (section 3.3.1). This is done to all 6 emotions, and we select the one that has the highest value. Additionally, if this value is below a certain threshold, then the emotion selected is neutral.

6.1 avgAU tests

We start by performing experiments with the AverageAU method. We experiment with various combinations of AU that are related to emotions, and also change the threshold of the method.

In table 6.1 we present the combinations of the action units for each study that we used. Hussain et al. [59] is a study worth mention since it summarized the works from the three last rows.

Method	Anger	Disgust	Fear	Happy	Sad	Surprise
Tautkute et al. (2019) [60]	4, 5, 7, 23	9, 15, 16	1, 2, 4, 5, 7, 20, 26	6, 12	1, 4 15	1, 2, 5, 26
Ghayoumi et al. (2016) [40]	2, 4, 7, 9, 10, 20, 26	2, 4, 9, 15, 17	1, 2, 4, 5, 15, 20, 26	1, 6, 12, 14	1, 4, 15, 23	1, 2, 5, 15, 16, 20, 26
Ekman et al. (1976) [45]	4, 5, 7, 23	9, 15, 16	1, 2, 4, 5, 20, 26	6, 12	1, 4, 15	1, 2, 5, 26
Lucey et al. (2010) [57]	4, 5, 15, 17	1, 4, 15, 17	1, 4, 7, 20	6, 12, 25	1, 2, 4, 15, 17	1, 2, 5, 25, 27
Karthick et al. (2013) [58]	4, 5, 7, 23, 24	9, 17	1, 4, 5, 7	6, 12, 25	1, 4, 15, 17	1, 2, 5, 26, 27

Table 6.1: Correspondence of AUs to emotions

In table 6.2 we display the results of the avgAU method on the CK+ dataset, using various combinations of AU and different thresholds. The threshold is the value where a facial expression will be classified as neutral if the average of the AU is not above it (section 3.3.1).

After the analysis of each combination, we decided to try using, for each emotion, the combination that achieved the highest score, we called this MixedBest. The combination from Ghayoumi et al. [40] and Lucey et al. [57], overall performed the worst, with the rest of the combinations achieving accuracy above 70%. Thus, we focus our attention on these combinations.

All combinations performed poorly, in terms of detecting neutral faces, with a threshold of 0.5. Tautkute et al. [60] and Karthick et al. [58] achieved the lowest accuracy scores on detecting fear. MixedBest failed the most on detecting anger. Ekman et al. [45] achieved high accuracy with the lowest being on fear,

nonetheless, is the combination that offers the most balanced scores. It also achieved similar results with thresholds 1 and 0.8, with fear, anger, and neutral having the most accuracy changes.

Method	Threshold	Anger(%)	Disgust(%)	Fear(%)	Happy(%)	Sad(%)	Surprise(%)	Neutral(%)	Overall(%)
Ghayoumi et al. [40]	0.8	20.0	16.61	33.6	98.55	38.57	89.16	74.44	59.14
Lucey et al. [57]	0.8	6.67	0.0	44.0	99.7	7.14	90.84	68.89	52.78
Tautkute et al. [60]	0.8	37.33	82.71	8.0	100.0	63.57	87.95	74.44	73.64
	1	27.11	76.61	4.0	100.0	61.43	85.54	87.78	70.76
	0.5	48.44	85.72	15.2	100.0	67.14	90.6	34.44	75.05
Ekman et al. [45]	0.8	37.33	82.71	26.40	100.0	63.57	87.95	72.22	74.92
	1	27.11	76.61	19.2	100.0	61.43	85.54	87.77	71.93
	0.5	48.89	85.76	30.4	100.0	67.14	90.36	31.11	76.02
Karthick et al. [58]	0.8	24.89	85.08	11.2	100.0	68.57	80.72	75.56	71.25
	1	20.89	81.02	8.8	99.7	65.71	78.8	93.33	69.97
	0.5	25.33	88.47	12.0	100.0	74.29	82.41	53.33	71.68
mixedBest	0.8	15.11	83.05	52.0	100.0	68.57	92.53	67.78	75.23
	1	14.22	78.98	47.2	100.0	65.71	91.57	86.67	74.56
	0.5	15.56	86.44	52.8	100.0	74.29	94.70	22.22	74.50

Table 6.2: Variations thresholds to the best avgAU methods

Ekman et al. achieved the best results, so in the following experiments this is the combination we are using for the avgAU method.

6.2 Experiments on Emotion algorithms

To compare the three algorithms for emotion recognition we first analyse each on our dataset. After inspecting the dataset of Vizzy, we decided to analyse two representative cases: First, a girl, whose facial expressions are visible, and where we noticed that all algorithms had almost no issue in detecting emotions. Second, a boy with a beard where the algorithms have more issues in recognizing the correct emotion.

In table 6.3, we show the results for all three algorithms, for our experiment. We present examples for Data 1 (girl) and Data 7 (bearded boy) in Figure 6.1.

For this experiment, we decided first to evaluate the capability of the algorithm to detect happiness and surprise during SNV errors. While annotating the dataset we noticed that, when a SNV occurred, the participants either did not react or reacted with a smile and/or laughter. As such, we can evaluate if the algorithms are effective, by checking if they can detect happiness or surprise during the SNV.

For data 1, all methods achieved over 80% accuracy. AverageAU performed best, with an accuracy of 94.3%. For data 7, Efficient CNN [44] performed worst while AverageAU performed best, with 13.74% and 75.42% accuracy, respectively. Since Efficient CNN [44] was unable to deal with the bearded example (Figure 6.1(b)) and was outperformed by all other algorithms, we argue that it is not suitable for our application.

To test the algorithms in other conditions we added the cases of TF and No Error. We noticed that generally when the TF error occurred (Vizzy arm destroying the construction), participants laugh, smile, or stare at Vizzy. As such, we assume that the most relevant emotions during this mistake, are happiness,

surprise, and neutral. When an error has not occurred (No Error cases), the participants were generally with a neutral face.

Again, the AverageAU method outperforms the other algorithms. We also noticed that when the no error situation was taken into consideration, the accuracy of DeepFace lowered considerably, meaning that it has a lower performance in identifying other emotions than happiness.

Data	SNV	TF	No Error	Efficient CNN(%)	DeepFace(%)	avg AU(%)
1	Happy, Surprise			83.5	88.35	94.3
7	Happy, Surprise			13.74	54.7	75.42
1,7	Happy, Surprise			50.2	72	85
1	Happy, Surprise		Neutral	53.65	37.05	60.7
7	Happy, Surprise		Neutral	46.32	51.7	58.37
1.7	Happy, Surprise		Neutral	50.23	43.89	59.63
1	Happy, Surprise	Happy, Surprise, Neutral		86.19	88.04	88.5
1,7	Happy, Surprise	Happy, Surprise, Neutral		66.85	81.06	84.34
1	Happy, Surprise	Happy, Surprise, Neutral	Neutral	57.58	43.02	63.75
1,7	Happy, Surprise	Happy, Surprise, Neutral	Neutral	52.49	47.71	61.87

Table 6.3: Experiments on Vizzy dataset

(a) Data1 sample **(b)** Data7 sample

Figure 6.1: Vizzy dataset images

To have a better comparison between DeepFace and the AverageAU method, we tested them on the FER2013 dataset[1] and the CK+ dataset [61].

In table 6.4 we present the results. DeepFace achieved 52% on FER2013, unfortunately, we could not use the AverageAU method on this dataset because the images provided were 48x48 resolution and openFace could not obtain the AU from the images. On the CK+ dataset, DeepFace achieved overall 31.0% accuracy, while the average AU achieved 74.92%. We decided to look at how each method performs for each emotion. Like we noticed in the previous experiment, DeepFace performs efficiently when detecting happiness, achieving 83.48%, however, on the rest of the emotions the performance was

[1] https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge/data

low. AverageAU outperformed DeepFace in all categories, having 100% accuracy on the happy emotion and the lowest 26.4% on fear emotion. In Figure 6.2 we show the confusion matrix of the avgAU method, we can see now in more detail that anger cases are classified as neutral, as well as disgust and sad. The sad cases are confused with disgust emotion and neutral. The fear case is the one where it fails the most.

We also used the McNemar test [62] to check the statistical significance of the disagreement between the two methods. McNemar test determined that there was a statistically significant difference between the avgAU and deepFace, $p < 0.01$.

Method	Dataset	Anger(%)	Disgust(%)	Fear(%)	Happy(%)	Sad(%)	Surprise(%)	Neutral(%)	Overall(%)
DeepFace	FER2013	40.02	46.36	38.91	69.66	42.38	65.06	51.3	52.53
DeepFace	CK+	23.11	6.4	8.8	83.48	21.43	12.29	62.22	31.0
avgAU	CK+	37.33	82.71	26.4	100.0	63.57	87.95	72.22	74.92

Table 6.4: Experiments on FER2013 and CK+ dataset

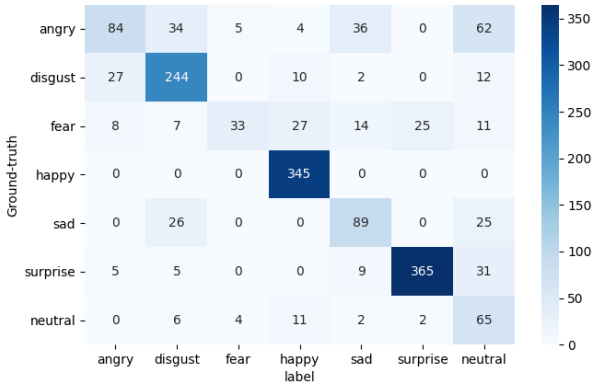


Figure 6.2: Confusion matrix of avgAU on CK+ dataset

We can see that DeepFace is good at detecting happiness and neutral emotions, but it fails in other emotions. The proposed emotion recognition algorithm AverageAU (avgAU) surprised us with its results:(i) It outperformed the other methods; (ii) It is computationally faster since it is just a calculation and does not require any data for training; (iii) And reduces the uncertainty of the results by only depending on the results of openFace, instead of another machine learning algorithm.

6.3 Fine-tuning AverageAU

Since Ekman et al. achieved the best results, we will fine-tune the algorithm, on an experiment similar to the one done in table 6.3, by checking the threshold that performs the best, table 6.5.

Data	SNV	TF	No Error	T0.8(%)	T0.5(%)	T1(%)	T1.5(%)
1	Happy, Surprise			94.3	94.3	94.3	90.11
7	Happy, Surprise			75.42	79.76	72.05	48.67
1,7	Happy, Surprise			85	87.35	83.68	70.34
1	Happy, Surprise		Neutral	60.7	39.65	72.42	94.0
7	Happy, Surprise		Neutral	58.37	44.53	65.9	82.39
1.7	Happy, Surprise		Neutral	59.63	41.93	69.38	
1	Happy, Surprise	Happy, Surprise, Neutral		88.5	78.99	95.3	95.3
1,7	Happy, Surprise	Happy, Surprise, Neutral		84.34	76.67	90.59	88.73
1	Happy, Surprise	Happy, Surprise, Neutral	Neutral	63.75	43.62	75.15	
1,7	Happy, Surprise	Happy, Surprise, Neutral	Neutral	61.87	44.66	71.64	89.4

Table 6.5: Ekman avgAU method

Thresholds 0.5 and 1.5 are excluded, they either remove important emotions or are too sensitive to a change in the AU and do not detect neutral faces. As such, we choose values between these two, especially 0.8 and 1. To choose which threshold to use we will first, for each interaction, run a similar test to the one presented, i.e. analyse how the method deals in identifying the emotions during SNV, TF and when there is no error. For instance, on data 1 we will choose threshold 1 since it does not reduce the accuracy during SNV but increases when there is no error.

6.4 AverageAU on faces with half masks

With the current pandemic going on, people now need to use masks to protect themselves and the people surrounding them. In this section, we perform tests to see if it is possible with the avgAU method to capture emotions using only the upper part of the face. In table 6.6 we present the action units from the Ekman correspondence that can be used.

Method	Anger	Disgust	Fear	Happy	Sad	Surprise
Ekman	4, 5, 7	9	1, 2, 4, 5	6	1, 4	1, 2, 5

Table 6.6: Ekman AU from superior part of the face (covid Ekman)

In table 6.7 we show the accuracy and on Figure 6.3 the confusion matrix. It is hard to identify fear with only the upper part of the face, however, the algorithm was able to identify the rest of the emotions. This shows that having the AU of the upper part of the face, the one that the mask does not hide, it is possible to identify emotions.

Method	Anger(%)	Disgust(%)	Fear(%)	Happy(%)	Sad(%)	Surprise(%)	Neutral(%)	overall(%)
covidEkman	36.0	94.58	0.0	89.28	52.86	83.13	88.88	71.38

Table 6.7: Results of covid Ekman on CK+ dataset

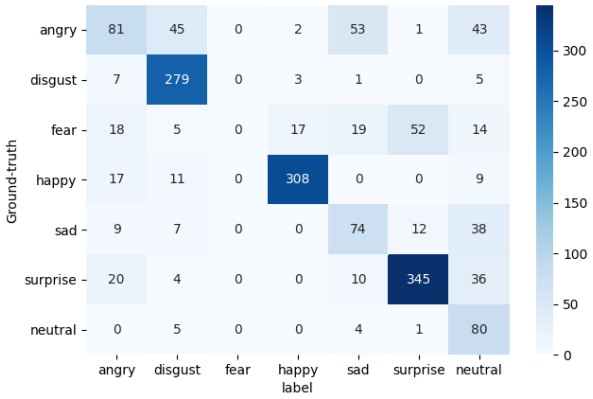


Figure 6.3: Confusion Matrix for covid Ekman

6.4.1 openFace and People with Masks

The previous action units were obtained with the whole face available to openFace, in this next experiment, we see how openFace behaves with faces with masks and check if the AU from the upper face continues to be calculated correctly. For these experiments, we will use images from the CK+ dataset and draw a mask on top of them for a direct comparison.

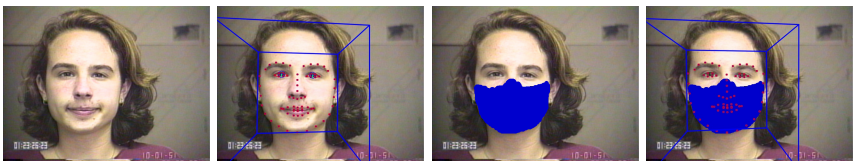
In Figures 6.4 we show an image from the CK+ dataset representing the neutral emotion, we also show the original image, 6.4(a), the image with a mask drawn, 6.4(c), and the bounding boxes placed by openFace, 6.4(b) 6.4(d). We also show the action units obtained for each image for the Ekman covid and the calculations to reach the emotion, table 6.8.

In Figure 6.5 we show representative images for each emotion, with and without a mask. OpenFace was able to detect the face even with the mask drawn. On the legend of the figures, we show the results of the avgAU method to obtain emotions, for the Ekman correspondence of specific AU to emotions and the Ekman covid.

We were expecting that the results from the original image with the Ekman covid were the same as the image with a mask using Ekman covid, however that was not the case, showing that the calculated upper AU with the mask is different from the upper AU without the mask.

AU	Mask	No Mask	Emotion	AU Mask	AU No Mask
01	0.0	0.0	Anger = $(AU04 + AU05 + AU07) \div 3$	0.1967	0.2867
02	0.0	0.0	Sad = $(AU01 + AU04) \div 2$	0.0	0.0
04	0.0	0.0	Happy = AU06	1.04	0.18
05	0.0	0.0	Fear = $(AU01 + AU02 + AU04 + AU05) \div 4$	0.0	0.0
06	1.04	0.18	Surprise = $(AU01 + AU02 + AU05) \div 3$	0.0	0.0
07	0.59	0.86	Disgust = AU09	1.75	0.0
09	1.75	0.0	Highest (Neutral < 0.8)	1.75	0.2867
Decision				Disgust	Neutral

Table 6.8: Action Units on Neutral image



(a) **Neutral**, Ekman: **Neutral**; Ekman covid: **Neutral**

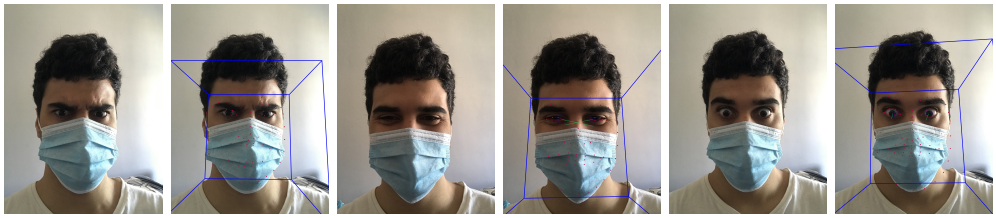
(b) Neutral Bounding box

(c) **Neutral**, Ekman covid: **Disgust**

(d) Neutral mask Bounding box

Figure 6.4: covid test on CK+ dataset, bounding box

We also tested openFace with images of a person using a real face mask, Figures 6.6(a), 6.6(c) and 6.6(e). OpenFace was able to detect the face, Figures 6.6(b), 6.6(d) and 6.6(f), and the avgAU method was able to correctly identify two emotions, the happy and the surprise.



(a) **Anger**; Ekman covid: **Happy**

(b) Anger bounding box

(c) **Happy**, Ekman covid: **Happy**

(d) Happy bounding box

(e) **Surprise**, Ekman covid: **Surprise**

(f) Surprise bounding box

Figure 6.6: covid test, with real face mask

Using an algorithm[2] to automatically mask the faces from the CK+ dataset, we tested how our emotion detector behaves in the entire dataset with masks, table 6.9 and Figure 6.7.

Method	Anger(%)	Disgust(%)	Fear(%)	Happy(%)	Sad(%)	Surprise(%)	Neutral(%)	overall(%)
Ekman Covid	1.5	100.0	0.0	0.0	0.0	5.07	0.0	20.6

Table 6.9: CK+ masked, avgAU result

[2] https://github.com/Prodesire/face-mask

(a) **Happy**; Ekman Covid:**Disgust**

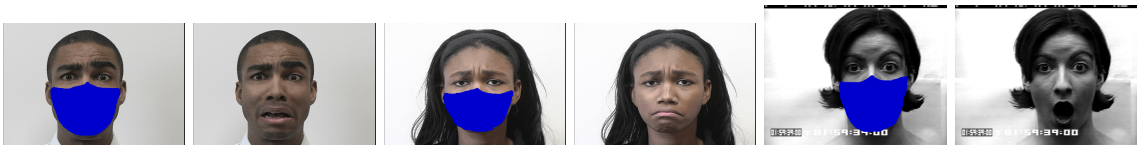
(b) **Happy**; Ekman: **Happy**; Ekman Covid:**Happy**

(c) **Anger**; Ekman covid: **Disgust**

(d) **Anger**; Ekman: **Anger**; Ekman covid: **Sad**

(e) **Disgust**; Ekman Covid: **Disgust**

(f) **Disgust**; Ekman: **Disgust**; Ekman Covid: **Disgust**



(g) **Fear**; Ekman covid: **Anger**

(h) **Fear**; Ekman: **Fear**; Ekman covid: **Surprise**

(i) **Sad**; Ekman covid: **Anger**

(j) **Sad**; Ekman: **Sad**; Ekman covid: **Sad**

(k) **Surprise**; Ekman covid: **Disgust**

(l) **Surprise**; Ekman: **Surprise**; Ekman covid: **Surprise**

Figure 6.5: covid test on CK+ dataset

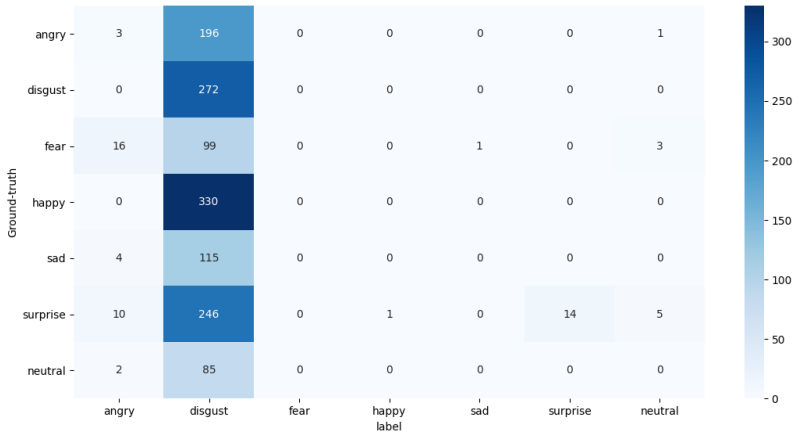


Figure 6.7: Confusion Matrix for covid Ekman, masked CK+

A reason for the action units being different is that OpenFace uses specific facial landmarks to align the face and to normalize it to compare to a neutral position [2]. The masks hide some of those landmarks, Figure 6.8. The facial landmarks are also used to obtain geometry and appearance face features, to then classify action units.

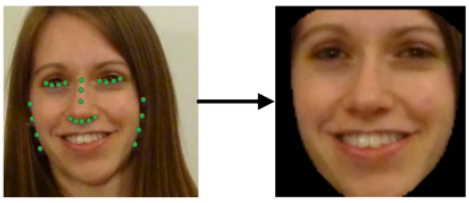


Fig. 2: Example of stable points used for alignment of the face to a common reference frame, followed by masking.

Figure 6.8: Figure from Zadeh et al. [2]

These experiments show that openFace results AU differently when people are using half masks. The results in table 6.7 show that the usage of only the upper action units are capable of detecting emotions, as such this algorithm has much potential and it is worth for future work to make an algorithm that can obtain AU from people that are using face masks.

7

Conclusion

In this work, we proposed an algorithm that detects and classifies error situations during one-to-one human-robot interactions in a controlled environment. The proposed pipeline uses facial and head features extracted from image frames of a robot onboard camera and information of robot actions. The proposed pipeline achieved significantly higher results when using the proposed set of features, which includes head, gaze, AU, emotions, and actions of the robot, than with features used in past works, that used head and gaze features [18, 19]. With an average accuracy of 72.77%, our algorithm showed promising results in the evaluation dataset. The usage of a median filter showed an improvement in the performance of the algorithm, with an average accuracy of 79.63%. Further tests validated the use of Random Forest models to detect errors and classify them with the proposed set of features. These results are obtained from an exhaustive study of the combination of several input features and classification algorithms. We want to stress the following results from the components of the pipeline:

- Random Forest classifiers work better on both error detection and error classification;
- Action units and robot context improve in a significant manner the performance of both error detection and error classification;
- Emotion features improve the performance of error detection but not error classification;
- The emotion recognition algorithm proposed in this work outperforms state-of-the-art methods in the case of our dataset. In addition, our method is computationally efficient when compared to deep learning-based methods.

We obtained promising results using the Isolation Forest algorithm, which is able to cope with mislabeled data while having similar performance to the conventional Random Forest. Future works should study these findings in detail. In future works, we intend to perform actual human-robot interaction studies to test our algorithm in real-time, making the robot react to error information. Moreover, we will explore more contextual information, for instance age or culture, as well as temporal image and action features. Finally, we also intend to evaluate the proposed emotion recognition algorithm in more challenging scenarios, such as dealing with multiple participants simultaneously.

Bibliography

[1] J. Avelino, A. Gonçalves, R. Ventura, L. Garcia-Marques, and A. Bernardino, "Collecting social signals in constructive and destructive events during human-robot collaborative tasks," in *Companion of the 2020 ACM/IEEE International Conference on Human-Robot Interaction*, 2020, pp. 107–109.

[2] A. Zadeh, Y. Chong Lim, T. Baltrusaitis, and L.-P. Morency, "Convolutional experts constrained local model for 3d facial landmark detection," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2017, pp. 2519–2528.

[3] T. Belpaeme, J. Kennedy, A. Ramachandran, B. Scassellati, and F. Tanaka, "Social robots for education: A review. science robotics 3, 21 (2018)," 2018.

[4] A. Sauppé and B. Mutlu, "The social impact of a robot co-worker in industrial settings," in *Proceedings of the 33rd annual ACM conference on human factors in computing systems*, 2015, pp. 3613–3622.

[5] J. Avelino, H. Simão, R. Ribeiro, P. Moreno, R. Figueiredo, N. Duarte, R. Nunes, A. Bernardino, M. Čaić, D. Mahr *et al.*, "Experiments with vizzy as a coach for elderly exercise," in *Proc. Workshop Pers. Robots Exercising Coaching-HRI Conf.(PREC)*, 2018, pp. 1–6.

[6] M. Giuliani, N. Mirnig, G. Stollnberger, S. Stadler, R. Buchner, and M. Tscheligi, "Systematic analysis of video data from different human–robot interaction studies: a categorization of social signals during error situations," *Frontiers in Psychology*, vol. 6, jul 2015.

[7] M. Salem, G. Lakatos, F. Amirabdollahian, and K. Dautenhahn, "Would you trust a (faulty) robot?" in *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction - HRI '15*. ACM Press, 2015.

[8] H. Yasuda and M. Matsumoto, "Psychological impact on human when a robot makes mistakes," in *Proceedings of the 2013 IEEE/SICE International Symposium on System Integration*. IEEE, dec 2013.

[9] C. G. Morales, E. J. Carter, X. Z. Tan, and A. Steinfeld, "Interaction needs and opportunities for failing robots," in *Proceedings of the 2019 on Designing Interactive Systems Conference*. ACM, jun 2019.

[10] I. Poggi and D. Francesca, "Cognitive modelling of human social signals," in *Proceedings of the 2nd international workshop on Social signal processing*, 2010, pp. 21–26.

[11] N. C. Krämer, A. von der Pütten, and S. Eimler, "Human-agent and human-robot interaction theory: similarities to and differences from human-human interaction," in *Human-computer interaction: The agency perspective*. Springer, 2012, pp. 215–240.

[12] P. Ekman, W. V. Friesen, and J. C. Hager, "Facial action coding system: The manual on cd rom," *A Human Face, Salt Lake City*, pp. 77–254, 2002.

[13] Y.-I. Tian, T. Kanade, and J. F. Cohn, "Recognizing action units for facial expression analysis," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 23, no. 2, pp. 97–115, 2001.

[14] C. Marinetti, P. Moore, P. Lucas, and B. Parkinson, "Emotions in social interactions: Unfolding emotional experience," in *Emotion-oriented systems*. Springer, 2011, pp. 31–46.

[15] A. Rausch, J. Seifried, and C. Harteis, "Emotions, coping and learning in error situations in the workplace," *Journal of Workplace Learning*, 2017.

[16] M. Tulis, "Error management behavior in classrooms: Teachers' responses to student mistakes," *Teaching and Teacher Education*, vol. 33, pp. 56–68, 2013.

[17] D. Kontogiorgos, S. van Waveren, O. Wallberg, A. Pereira, I. Leite, and J. Gustafson, "Embodiment effects in interactions with failing robots," in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. ACM, apr 2020.

[18] D. Kontogiorgos, A. Pereira, B. Sahindal, S. van Waveren, and J. Gustafson, "Behavioural responses to robot conversational failures," in *Proceedings of the 2020 ACM/IEEE International Conference on Human-Robot Interaction*. ACM, mar 2020.

[19] P. Trung, M. Giuliani, M. Miksch, G. Stollnberger, S. Stadler, N. Mirnig, and M. Tscheligi, "Head and shoulders: automatic error detection in human-robot interaction," in *Proceedings of the 19th ACM International Conference on Multimodal Interaction - ICMI 2017*. ACM Press, 2017.

[20] P. Moreno, R. Nunes, R. Figueiredo, R. Ferreira, A. Bernardino, J. Santos-Victor, R. Beira, L. Vargas, D. Aragão, and M. Aragão, "Vizzy: A humanoid on wheels for assistive robotics," in *Advances in Intelligent Systems and Computing*. Springer International Publishing, dec 2015, pp. 17–28.

[21] N. Mirnig, G. Stollnberger, M. Miksch, S. Stadler, M. Giuliani, and M. Tscheligi, "To err is robot: How humans assess and act toward an erroneous social robot," *Frontiers in Robotics and AI*, vol. 4, may 2017.

[22] R. Gehle, K. Pitsch, T. Dankert, and S. Wrede, "Trouble-based group dynamics in real-world HRI &#x2014 reactions on unexpected next moves of a museum guide robot," in *2015 24th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*. IEEE, aug 2015.

[23] C. J. Hayes, M. Moosaei, and L. D. Riek, "Exploring implicit human responses to robot mistakes in a learning from demonstration task," in *2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*. IEEE, aug 2016.

[24] M. Stiber and C.-M. Huang, "Not all errors are created equal: Exploring human responses to robot errors with varying severity," 2020.

[25] N. Mirnig, M. Giuliani, G. Stollnberger, S. Stadler, R. Buchner, and M. Tscheligi, "Impact of robot actions on social signals and reaction times in HRI error situations," in *Social Robotics*. Springer International Publishing, 2015, pp. 461–471.

[26] D. E. Cahya, R. Ramakrishnan, and M. Giuliani, "Static and temporal differences in social signals between error-free and erroneous situations in human-robot collaboration," in *Social Robotics*. Springer International Publishing, 2019, pp. 189–199.

[27] T. Baltrusaitis, A. Zadeh, Y. C. Lim, and L.-P. Morency, "Openface 2.0: Facial behavior analysis toolkit," in *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*. IEEE, 2018, pp. 59–66.

[28] E. Krahmer, M. Swerts, M. Theune, and M. Weegels, "Error detection in spoken human-machine interaction," *International journal of speech technology*, vol. 4, no. 1, pp. 19–30, 2001.

[29] P. Barkhuysen, E. Krahmer, and M. Swerts, "Problem detection in human–machine interactions based on facial expressions of users," *Speech Communication*, vol. 45, no. 3, pp. 343–359, mar 2005.

[30] S. Ehrlich and G. Cheng, "A neuro-based method for detecting context-dependent erroneous robot action," in *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*. IEEE, nov 2016.

[31] S. K. Kim, E. A. Kirchner, A. Stefes, and F. Kirchner, "Intrinsic interactive reinforcement learning – using error-related potentials for real world human-robot interaction," *Scientific Reports*, vol. 7, no. 1, dec 2017.

[32] P. Ekman, "An argument for basic emotions," *Cognition & emotion*, vol. 6, no. 3-4, pp. 169–200, 1992.

[33] S. Li and W. Deng, "Deep facial expression recognition: A survey," *IEEE Transactions on Affective Computing*, pp. 1–1, 2020.

[34] C. A. Corneanu, M. O. Simon, J. F. Cohn, and S. E. Guerrero, "Survey on RGB, 3d, thermal, and multimodal approaches for facial expression recognition: History, trends, and affect-related applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 8, pp. 1548–1568, aug 2016.

[35] Z. Zhang, P. Luo, C. C. Loy, and X. Tang, "From facial expression recognition to interpersonal relation prediction," *International Journal of Computer Vision*, vol. 126, no. 5, pp. 550–569, nov 2017.

[36] J. V. den Stock, R. Righart, and B. de Gelder, "Body expressions influence recognition of emotions in the face and voice." *Emotion*, vol. 7, no. 3, pp. 487–494, aug 2007.

[37] F. Noroozi, D. Kaminska, C. Corneanu, T. Sapinski, S. Escalera, and G. Anbarjafari, "Survey on emotional body gesture recognition," *IEEE Transactions on Affective Computing*, pp. 1–1, 2019.

[38] R. Kosti, J. M. Alvarez, A. Recasens, and A. Lapedriza, "Emotion recognition in context," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jul 2017.

[39] M. Wegrzyn, M. Vogt, B. Kireclioglu, J. Schneider, and J. Kissler, "Mapping the emotional face. how individual face parts contribute to successful emotion recognition," *PloS one*, vol. 12, no. 5, p. e0177239, 2017.

[40] M. Ghayoumi and A. K. Bansal, "Unifying geometric features and facial action units for improved performance of facial expression analysis," *CoRR*, vol. abs/1606.00822, 2016. [Online]. Available: http://arxiv.org/abs/1606.00822

[41] S. Velusamy, H. Kannan, B. Anand, A. Sharma, and B. Navathe, "A method to infer emotions from facial action units," in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011, pp. 2028–2031.

[42] A. Saxena, , A. Khanna, and D. Gupta, "Emotion recognition and detection methods: A comprehensive survey," *Journal of Artificial Intelligence and Systems*, vol. 2, no. 1, pp. 53–79, 2020.

[43] S. I. Serengil and A. Ozpinar, "Lightface: A hybrid deep face recognition framework," in *2020 Innovations in Intelligent Systems and Applications Conference (ASYU)*. IEEE, 2020, pp. 23–27.

[44] H. Siqueira, S. Magg, and S. Wermter, "Efficient facial feature learning with wide ensemble-based convolutional neural networks," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 04, 2020, pp. 5800–5809.

[45] P. Ekman and W. V. Friesen, "Measuring facial movement," *Environmental Psychology and Nonverbal Behavior*, vol. 1, no. 1, pp. 56–75, 1976. [Online]. Available: https://doi.org/10.1007%2Fbf01115465

[46] F. Wilcoxon, "Individual comparisons by ranking methods," in *Breakthroughs in statistics*. Springer, 1992, pp. 196–202.

[47] Student, "The probable error of a mean," *Biometrika*, pp. 1–25, 1908.

[48] S. S. Shapiro and M. B. Wilk, "An analysis of variance test for normality (complete samples)," *Biometrika*, vol. 52, no. 3/4, pp. 591–611, 1965.

[49] P. D. Bridge and S. S. Sawilowsky, "Increasing physicians' awareness of the impact of statistics on research outcomes: comparative power of the t-test and wilcoxon rank-sum test in small samples applied research," *Journal of clinical epidemiology*, vol. 52, no. 3, pp. 229–235, 1999.

[50] J. Cohen, *Statistical power analysis for the behavioral sciences*. Academic press, 2013.

[51] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation-based anomaly detection," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 6, no. 1, pp. 1–39, 2012.

[52] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: identifying density-based local outliers," in *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, 2000, pp. 93–104.

[53] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural computation*, vol. 13, no. 7, pp. 1443–1471, 2001.

[54] P. J. Rousseeuw and K. V. Driessen, "A fast algorithm for the minimum covariance determinant estimator," *Technometrics*, vol. 41, no. 3, pp. 212–223, 1999.

[55] R. Domingues, M. Filippone, P. Michiardi, and J. Zouaoui, "A comparative evaluation of outlier detection algorithms: Experiments and analyses," *Pattern Recognition*, vol. 74, pp. 406–421, 2018.

[56] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification," *Machine learning*, vol. 85, no. 3, p. 333, 2011.

[57] P. Lucey, J. F. 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 *2010 ieee computer society conference on computer vision and pattern recognition-workshops*. IEEE, 2010, pp. 94–101.

[58] K. Karthick and J. Jasmine, "Survey of advanced facial feature tracking and facial expression recognition," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 2, no. 10, pp. 2278–1021, 2013.

[59] N. Hussain, H. Ujir, I. Hipiny, and J. Minoi, "3d facial action units recognition for emotional expression," *CoRR*, vol. abs/1712.00195, 2017. [Online]. Available: http://arxiv.org/abs/1712.00195

[60] I. Tautkutė and T. Trzciński, "Classifying and visualizing emotions with emotional dan," *Fundamenta Informaticae*, vol. 168, no. 2-4, pp. 269–285, 2019.

[61] P. Lucey, J. F. 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 *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*, 2010, pp. 94–101.

[62] Q. McNemar, "Note on the sampling error of the difference between correlated proportions or percentages," *Psychometrika*, vol. 12, no. 2, pp. 153–157, 1947.