Università Politecnica delle Marche
Scuola di Dottorato di Ricerca in Scienze dell'Ingegneria
Corso di Dottorato in Ingegneria Industriale

Research and development of a remote usability testing platform for better user eXperience

Ph.D. Dissertation of:

Abudukaiyoumu Talipu

Supervisor:

Prof. Maura Mengoni

Ph.D. Course coordinator:

Prof. Giovanni di Nicola

XXXIII edition - new series

Università Politecnica delle Marche
Scuola di Dottorato di Ricerca in Scienze dell'Ingegneria
Corso di Dottorato in Ingegneria Industriale

Research and development of a remote usability testing platform for better user eXperience

Ph.D. Dissertation of:

Abudukaiyoumu Talipu

Supervisor:

Prof. Maura Mengoni

Ph.D. Course coordinator:

Prof. Giovanni di Nicola

XXXIII edition - new series

Università Politecnica delle Marche
Dipartimento di (nome del dipartimento dove la tesi e' stata sviluppata)
Via Brecce Bianche — 60131 - Ancona, Italy

Acknowledgements

Foremost, I would like to express my sincere gratitude to my advisor Prof.Maura Mangoni for the continues support of my Ph.D study and research. Her guidance helped me in all the tie of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my Ph.D study.

Besides my advisor, I would like to thank FLOWING s.r.l and Antonio Dellava for helping me in the validation process of the remote usability platform and conducting the case study.

To all my friends and others who in one way or another shared their support, either morally, financially or physically, thank you.

Last but not least, I am greatly thankful to my family for always supporting me spiritually throughout my life.

Abstract

Nowadays smartphones and laptops equipped with cameras have become an integral part of our daily lives. The pervasive use of cameras enables the collection of an enormous amount of data, which can be easily extracted through video images processing. This opens up the possibility of using technologies that until now had been restricted to laboratories, such as gaze tracking and emotion analysis systems, to analyze users' behavior during the interaction with websites.

By implementing deep learning algorithms, face detection, facial expressions recognition and gaze tracking convolutional neural network models can be trained and implemented in remote usability testing. The thesis studies the traditional usability testing, modern day usability testing methods and proposes a more advanced deep learning based remote usability testing platform. The development process of the low-cost platform, training of the deep learning models, gaze tracking dataset building, finally the evaluation of the platform by conducting a case study are explained in detail. Some aspects of UX assessment that could help in addressing the usability such as user insights, engagement is explored and the development of a dashboard for the usability testing platform to present the automatically analyzed results is described.

The data obtained using the usability testing platform is particularity important in revealing more user insights and engagements as well as addressing effectiveness, efficiency and satisfaction aspects of usability testing.

Contents

List of Figures

List of Tables

Chapter 1. Introduction

1.1 User Experience and Usability

This thesis studies the User Experience and Usability aspects of web applications. A fully automated remote usability testing platform based on machine learning to monitor user engagement, satisfaction is also developed. It will assist User Interface (UI) /User Experience (UX) designers and developers to quickly conduct usability tests and obtain results immediately in almost everywhere.

User Experience (UX) is a domain in the field of Human Computer Interaction (HCI). It is the experience the software product or application creates for the people who use it in the real world (Garrett 2010). UX extends far beyond the functionality aspects of the user interaction, it also includes emotions and attitudes of the user. As the involvement of emotions and attitudes of the user in the context of UX, it brings new challenges to the UX designers to correctly measure and make designing decisions. As complexity of technology grows, UX must be given more attention and importance, UX will become critical part of the development process to provide that's efficient, easy to use and engaging (Kaufmann 2016).

Usability usually considered as a subdomain of UX that reflects the efficiency and "easy to use" aspects of it. It is defined as the ability of the user to use the application to carry out a specific task successfully (Albert 2013). The activities focus on measuring the usability of a product is called usability testing. Usability testing is a way to gather valuable feedbacks from the representative users, which can help designers and developers to make the product more appealing, more usable and more relevant to their audiences.

Traditionally usability tests are conducted in laboratories. The people recruited are invited to come to the test room, where the participants are asked to accomplish specific tasks, an observation room usually connected to the test room with a one-way mirror and sophisticated video and audio recording facilities consists of the typical usability testing laboratory. The cost of setting up such laboratories, the travel cost of recruited participants, more importantly the time spent on conducting these types of usability tests are huge.

Over the advancement of technologies, most of the problems existed before are already gone. These days, usability testing can be conducted with a laptop and the software that people use every day. Moreover, remote usability testing in which the test conductor and

participants are not co-located but interact over a computer or telephone can be more cost effective (A.J. Bernheim Brush 2004). Several comparison studies of traditional and remote usability testing have found that “no reliable difference between lab-based and remote testing in terms of the usability issues found, their type and their severity” (Services 2006).
User-centered design and usability testing of some user interfaces is being used in some traditional software development processes (Scholtz 2001). By focusing on use and usability instead of features and functionality, the final system can be turned into a better tool for the job that is smaller, simpler and ultimately less expensive (Ambler 2004).
Thanks to the recent development of machine learning (ML) and artificial neural network (NN) algorithms, many challenging, even thought to be “impossible to solve” problems have found their solutions. Face detection, Age and Gender detection, Facial expression recognition, become more and more accurate, new applications based on these technologies are already available at the people’s fingertips.

1.2 Research Objectives

This thesis contributes to UX and Usability research by implementing state of the art DL techniques in remote usability testing field. DL based analytics and low-cost, efficient remote usability testing platform is developed to visualizing remote usability test results as well as user interaction and engagements.

1.3 Thesis Structure

The next chapter presents in depth literature study of UX, usability and remote usability testing. Chapter 3 describes the preliminary tests on AI-based platform development on desktop and mobile devices, the training process of different deep learning CNN models are also explained. Chapter 4 is about designing processes, architectural structure and important features of the platform, Chapter 5 is about the case study conducted on the developed platform. Analysis on manual and platform obtained results are compared, the effectiveness and validity of the platform is analyzed. In Chapter 6, the conclusion is drawn, results are discussed, and future work and improvements could be made are listed.

Chapter 2. Research Background

2.1 Usability

In the early days, before the term "User-Centric Design" (UCD) came into existence, the main focus was on the product not on the user. Many computer programs that are technically well-designed fail to meet the human or organizational purposes they were designed to serve (Kling 1977). People later realized that the goal of the product is defined by the user, therefor "User-Centric Design" became the main focus. Since the 80s, a wealth of theories, methods, and design guidelines have been developed in the field of HCI with the aim of making products easier to learn and use in the long run. The field of HCI was for a long time identified as the field of usability engineering (Law 2009).

Usability was coined in order to replace the term "user friendly" which by the early 1980s had acquired a host of undesirably vague and subjective connotations (Bevana 1991). Usability may have been less important when personal computer products first appeared. In the early stages of any developing technology, products differ widely in the functionality they offer (Dumas 1999). As competitors appear, products mature, differences in functionality are no longer sufficient to overcome poor usability. Usability then weights more, and the focus is on usability of the product. (Bevana 1991) relates

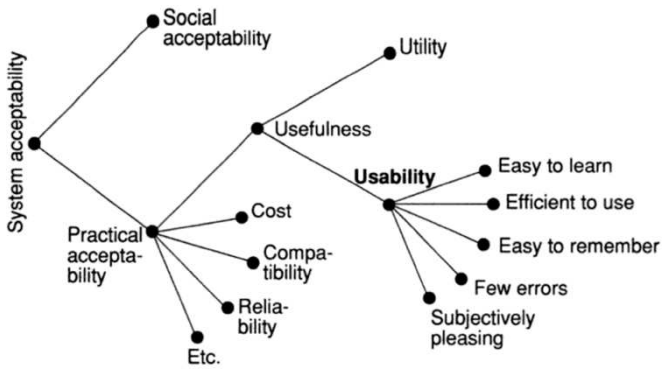


Figure 1 A model of the attributes of system acceptability (Nielsen 1994)

different approaches to usability and proposes a definition for it which states that usability should be defined as the ease of use and acceptability of a product for a particular class of users carrying out specific tasks in a specific environment.

The overall acceptability of a digital product or computer system is a combination of its social acceptability and its practical acceptability. Assuming a computer system is socially accepted, further analysis on practical acceptability which includes traditional categories such as cost, compatibility, reliability and others can be conducted. Usefulness is the issue of whether the system can be used to achieve some desired goal. It can be broken into two further categories utility and usability, where utility is the question of whether the functionality of the system is principle can do what is needed, and usability is the question of how well users can use that functionality (Nielsen 1994).
The ISO 9241-11 standard (2019) (Standardization 2018) defines usability as:

> *"the extent to which a product can be used by specific users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use"*

Everybody benefits from usability. Users obviously benefit from a product reduces the learning curve and allows them to do more with less effort. Companies benefit when the users buy more products, it helps companies to enhance their reputation, easy to use products will reduce product support, training costs. "Word of mouth" can also help companies reduce advertisement costs.

2.2 User Experience

The term "User Experience" was brought to wider attention by Apple (Norman 1995). A lot of studies have been conducted on explicitly defining what constitutes the UX. In (Law 2009), author gathered the views on UX of 275 researchers and practitioners from academia and industry. The paper concluded most respondents agree that UX is dynamic, context-dependent, and subjective which seems to be in line with the ISO definition of UX. The ISO 9241-210 (2019) (Standardization 2018) defines user experience as:

> *"a person's perceptions and responses that result from the use or anticipated use of a product, system or service"*

At a glance in the literature on UX reveals three major perspective. One thread predominantly deals with addressing human needs beyond the instrumental; a second thread stresses affective and emotional aspects of the interaction; and a third thread deals with the nature of experience. (Hassenzahl 2006).

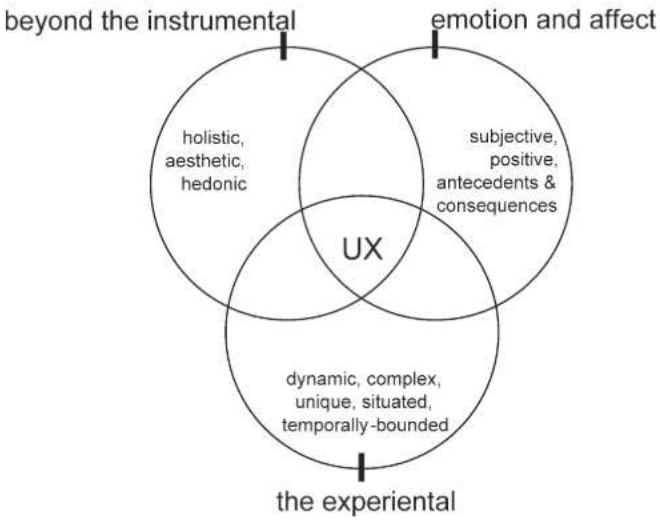


Figure 2 Facets of UX

The thread stresses affective emotional aspects of the interaction is the main focus in this context. The concept of affect refers to a large variety of psychological states such as emotions, feelings moods, sentiments and passions. Each of these affective states varies duration, impact and eliciting conditions of states, emotions are most relevant for product experience because only they imply one-to-one relationship between affective state and a particular object (William S. Green 2002).

Affective computing is a relatively new research domain defined in 1995 as computing that relates to, arises from, or deliberately influences emotion or other affective phenomena (Picard 1995). It is a part of human-computer interaction research which focuses on creating systems that take into account users' emotional states. Affective computing takes a 'computer' perspective. It predominantly deals with questions such as how computers can sense user affect, adapt to it, or even express its own affective response (Picard and Klein 2002).

Although UX research shares Affective Computing's recognition of affect and emotions, it is rather concerned with affective consequences on the human side than with technology, which is able to have affect (Hollnagel 2003). UX takes a 'human' perspective (Hassenzahl and Tractinsky 2006).

Satisfaction is the hardest to measure, as it is defined as the comfort and acceptability of the work system to its users (Landowska 2015). The most frequently applied methods are based on cognition (a user is asked about the perceived usability aspects) and behavioral metrics (interaction with application is monitored and then tagged with

diverse events and behaviors) (Nielsen 1994). Behavioral metrics provide criteria to evaluate effectiveness and efficiency, but not satisfaction factor. While applied in emotion evaluation, interviews and questionnaires (cognitive methods) might be significantly subjective, as people differ in perceived emotional states, that are considered important and allowed to be expressed (Landowska 2013).

The emotional aspects of customer behavior are attracting more and more interests in recent years. Experience with technology as characterized by McCarthy and Wright consists of a sensual, an emotional, a compositional and a spatio-temporal thread (Moen 2007). Hence, accessing the emotional state of users is crucial for developing satisfying products that are rich in experience. (Ganglbauer et al. 2011).

There are a few studies on fusing affect recognition and usability evaluation. (Zimmermann and Gomez 2006) described a feasibility study of behavioral method for measuring user affect in parallel with task processing. (Lew et al. 2012) is an example of affect evaluation applied in quality assurance procedures for web application. Another important work is (Ahn and Picard 2014) which proposes the Affective-Behavioral Cognitive (ABC) framework for user experience evaluation. The framework was validated with an experiment on beverages. (Kolakowska et al. 2013) proposes application of affect/emotion recognition in usability evaluation and have proposed four scenarios: first impression test, task-based usability test, free interaction test and comparative test. In (Landowska 2015), affect recognition methods and emotion representation models are reviewed and evaluated for applicability in usability testing procedures. (Landowska 2013) provides a brief review of methods used for affect recognition, and it may be used to address affective aspect of e-education.

2.3 Usability Testing

Usability testing refers to evaluating a product or service by testing it with representative users. It is a way of assessing the degree to which an interactive system is easy and pleasant to use with a view of identifying usability problems and/or a collection of usability measures/metrics. Furthermore, these qualitative and quantitative measures can be used to determine participant's satisfaction. Typically, during the test, participants will try to finish some predefined tasks while observers watch, listen and take notes. Usability testing can be carried out in a usability laboratory or online. From the test location point of view, lab-based (traditional) usability testing and online (remote) usability testing exist.

Traditional usability testing is conducted in dedicated laboratories implementing the methods of experimental design. The laboratory is equipped with one-way glass and

systems enabling facilitators to interact with and observe the tester, recording their on-screen actions, facial expressions, and verbal feedback. This approach may be more precise, but there are costs associated with hiring or owning a laboratory, recruiting testers and sending staff to facilitate and observe the testing (Gardner 2007). The investment to make this happen can be small or large, depending on the size and complexity of the space and the equipment in it (CAROL 2020).
Lab configurations can be one room, two rooms, three rooms, or even four rooms. One-room labs place the participant, the test team, and observers together. Two-room labs have one room for the participant and another room, often called the control room, for the test team and observers. Three-room labs have a separate room for observers, often called the executive viewing room. Four-room labs can be configured to have a separate room for focus groups or other types of user research (CAROL 2020).
Usually, organizations rated mature in UX studies are twice as likely to have a dedicated usability laboratory as those rated as nonmature.
In order to maximize the effectiveness of the laboratory-based usability tests, the lab environment should be as realistic as time and budget allow.

Figure 3 Two-room usability lab with one-way mirror

Traditional usability testing methods are difficult to use in producing web sites and web applications mainly because of the decreased development times that companies demand for this type of software. Companies want to use web sites to sell merchandise and provide services to customers. Therefore, it is essential to make usability a high priority in the development of web-based software (Scholtz 2001).

Figure 4 Three-room usability lab shows the view from executive viewing room through the control room into the participant room

In remote usability testing, observers may use online screen sharing software and webcams to observer the participants actions and their facial expressions in real time or watch session recordings instead. These approaches known as remote moderate usability testing and remote unmoderated usability testing.

Each method has its advantages and disadvantages. A well-equipped laboratory provides a controlled environment that enables detailed recording and observation of the tester, making it easier to analyze results. However, laboratory facilities are expensive and create an unnatural environment for a user. Furthermore, travel to the test facility may be inconvenient and time-consuming for both testers and facilitators (Gardner 2007). Remote, rapid and automated usability testing tools are helpful in providing more usability information in a shorter time and in a form that can be immediately useful to usability professionals (Scholtz 2001).

Remote usability testing is becoming more and more popular specially when it comes to web usability testing. Remote usability testing conveniently allows both tester and facilitator to work from home or office. It will also increase the pool of potential participants if the users are decentralized. This is particularly valuable for international companies. A further advantage of remote usability is the opportunity to observe the web application on a variety of computer configurations, including different operating systems, screen resolutions, and internet connection speeds (Gardner 2007). At the same time, this could cause some unexpected failures when the system is incompatible with

the participants computer settings. The time for page response during the remote tests is one of the important characters.

Whether it is traditional usability test or remote usability test, test tasks or scenarios need to be representative of typical tasks conducted by most users. Usability testing helps in discovering mistakes committed by users when interacting with system's interfaces (F. M. EL-firjani, K. Elberkawi, and M. Maatuk 2017). The selection of the users that truly represent the entire user population in accomplishing given testing tasks is needed. During a usability test, a target user population should be selected and recruited (Lazar 2001).

After the completion of the tests, the process of turning the collected data, transcripts, and observations into an actionable report on usability issues can be started. One of the most resource demanding activities in a usability evaluation is the analysis of collected empirical data. Not only is it time consuming, but data analysis is also very vital as it extracts key findings of the usability evaluation. Furthermore, it is a key activity in usability evaluations as evaluators may find themselves influencing the findings through different interpretations (Kjeldskov, Skov, and Stage 2004). It is helpful to review the original goals to be evaluated, it makes the evaluators to stay focused on the most relevant feedback. Many methods and techniques exit for analyzing the empirical data from usability evaluations, for example, grounded analysis (Anselm Strauss 1997), video data analysis (Nayak and Mrazek 1994), cued-recall (Omodei 2002) and expert analysis (Nielsen and Molich 1990) etc. However, instrumentation in data analysis of usability evaluations is often poorly discussed (Gray and Salzaman 1998) and the relative value of applying these methods and techniques to analysis of usability is still largely unidentified. Of special interest, it seems implicitly assumed that thorough video analysis with detailed log-files and transcriptions of usability evaluation sessions is the most optimal way to analyze usability evaluation data (Kjeldskov, Skov, and Stage 2004). However the added value of spending large amount of time on video analysis in relation to the results subsequently produced is still questionable (Nielsen 1994).

In (Følstad, Law, and Hornbæk 2012) 155 usability practitioners are surveyed on the analysis in their latest usability evaluations and concluded analysis support from academic research, including tools, forms and structured formats, does not seem to have direct impact on analysis practice.

Usability evaluation methods in the literature are Cooperative evaluation (Følstad, Law, and Hornbæk 2012), Cooperative usability testing (Frøkjær and Hornbæk 2005), Contextual evaluation. A comparison study of three remote asynchronous usability testing methods are studied in (Bruun et al. 2009). They are compared to each other and to a classical laboratory-based approach. It is reported in overall, three remote methods

performed significantly below the classical lab test in terms of the number of usability problems identified. However, it also suggests the three remote methods seem to complement each other, thus combination of two or all three is a cost-effective solution. Test task design and specification are a core resource for many inspection and model-based approaches.

There are three important focus points when analyzing the usability test data:

- WHAT is the current critical task
- WHERE is the critical task is failing and by how much
- WHY the critical task is failing

Depending on at which point the usability testing is conducted, it can be subdivided into two types formative testing and summative testing. Formative testing refers the testing conducted when the product is still in development with the goal to diagnose and fix problems, typically based on small studies and repeated during development. Summative testing is when the product is nearly finished or finished, with a goal of establishing a baseline of metrics validating that the product meets requirements. Generally, it requires large numbers for statistical validity (CAROL 2020).

2.3.1 Measures of Usability Testing

As it is stated in the usability definition effectiveness, efficiency and satisfaction are the three critical measures of Usability. Usability testing should be conducted addressing those critical measures.

Effectiveness represents the accuracy and completeness with which users achieve certain goals and is typically measured through observed error rates, binary task completion, and quality of the outcome for a given task (Frøkjær 2000) (Hornbæk 2007). Efficiency can be characterized as effectiveness in relation to resources spent and is typically measured through task completion time and learning time (Law 2009). The third one, user satisfaction, represents users' comfort in using and overall attitude to the product and is typically measured through psychometric scales on overall preference, product quality perceptions and specific attitudes towards the interface (Hornbæk 2007).

Although measures of effectiveness and efficiency are, to some extent, determined by the user's perceptions of these qualities, there is no denying that the measure of satisfaction is derived wholly from the user's perception of satisfaction.

From the data collection point of view, upper mentioned measures can be classified into qualitative and quantitative methods. Qualitative data is of free-form and non-numerical, such as diaries, open-ended questionaries. Qualitative methods are more process oriented, involves using the observations or comments to come to a conclusion. Quantitative data on the other hand, can be coded in a numerical form. Quantitative

methods are specific, testable outcome-oriented, involves experiments or closed questions or rating scales.

2.4 Methods and Approaches

The selection of usability testing methods depends on the actual goal of the facilitator, whether they want to conduct a small-scale formative test or a large-scale summative test.

2.4.1 Formative test

Conducting small usability testing studies are commonly recognized as an effective means to understand user's goals, motivations, and engagement with the product. they can provide the development team with list of findings to analyze and fix, then conducting another small study to see whether the fixes actually worked.
Small studies don't provide large data to conduct detailed analysis, but they provide great insights to developers that can be put into action right away. Most of the time formative tests will produce qualitative data as a result. Standard small usability studies incorporate following essential elements (CAROL 2020):

- Establish the user profile
- Create task-based scenarios
- Use a think-aloud process
- Make changes and test again

Almost all of the products are designed to serve the needs of a wide variety of users with different skill levels, domain knowledge, and a host of other factors. Even within a clearly defined user group, there exists some significant variations. When conducing small formative tests, the number of participants is usually small. The participants should be the correct representations of the user population. Choosing the right subgroup of the user population is the most important part of the test.
For obtaining useful results, tasks should be created for the participants to perform during the test. These tasks should be realistic descriptions of some goals, the facilitator can observe the participants' methods for achieving those goals.
A think-aloud process is one of the methods in traditional usability testing in which participants are encouraged to share their thoughts while they conduct the tasks. With think-aloud method participants share their thoughts, reactions, pleasure, pain. It helps facilitator understand their experience better. It is worth mentioning that researchers who studied the impact of the think-aloud method on the accuracy of the metrics are divided.

Some indicates that think-aloud adversely affects measurements and other indicating that it has no adverse effect (Lewis, 2014).
At last, after fixing the detected problems, same tests could be conducted again to check whether the fixies actually improved the product. From the establishment of user profile to conducting the test can be carried out in iterative manner, which will improve the product over time. However, it also depends on the development budget and delivery timeline.

2.4.2 Summative test

Summative testing is when the product is nearly finished or finished, with a goal of establishing a baseline of metrics validating that the product meets requirements. Generally, it requires large numbers for statistical validity (CAROL 2020). The result of the test is generally used to produce metrics, such as average time on task, completion rates, error rates, optimal navigation path, and other measures.
Essential key elements for conducting large scale summative studies (CAROL 2020) :

- Using the same tasks/scenarios or different ones
- Gathering metrics to quantify results
- Choosing a testing method or combination of methods which includes:
  - Moderated
  - Unmoderated
- Balancing the goals, management, budget and time

By using same tasks/scenarios, the improvement in usability or UX can be measured similar to the formative testing. It can also focus on a specific feature and usability among the participants to uncover hidden flaws.
A metric is a way of measuring or evaluating a particular phenomenon or a thing. Different types of metrics can be obtained by conducting the test such as task completion, time spent on each task, error rate etc.
Choosing the right method or combination of methods highly depends on the user goal, the budget and time constrains. Fortunately, there exists different kinds of methods available for almost every situation.

2.4.3 Usability Metrics

Effectiveness, efficiency and satisfaction are the three critical aspects of Usability. Usability testing should be conducted based on these critical aspects.
Effectiveness represents the accuracy and completeness with which users achieve certain goals and is typically measured through observed error rates, binary task completion,

and quality of the outcome for a given task (Frøkjær 2000) (Hornbæk 2007). Efficiency can be characterized as effectiveness in relation to resources spent and is typically measured through task completion time and learning time (Law 2009). The third one, user satisfaction, represents users' comfort in using and overall attitude to the product and is typically measured through psychometric scales on overall preference, product quality perceptions and specific attitudes towards the interface (Hornbæk 2007).
Although measures of effectiveness and efficiency are, to some extent, determined by the user's perceptions of these qualities, there is no denying that the measure of satisfaction is derived wholly from the user's perception of satisfaction.
Over the time there are many published articles about user related metrics. It is important to study these metrics and what exactly they measure. In (Rodden, Hutchinson, and Fu 2010) published by google, a framework called HEART for user-centered metrics for web applications and how those metrics help their team make decisions both in data-driven and user-centered ways. For monitoring the overall health of a product, they proposed PULSE metrics stands for Page views, Uptime, Latency and Seven-day active users. Complimentary to PULSE, for addressing more UX quality HEART metrics is created which refers to Happiness, Engagement, Adoption, Retention, and Task success. GOALS-SIGNALS-METRICS explicitly relates to a goal and can be used to track progress towards that goal.
From the data collection point of view, upper mentioned measures can be classified into qualitative and quantitative methods. Qualitative data is of free-form and non-numerical, such as diaries, open-ended questionaries. Qualitative methods are more process oriented, involves using the observations or comments to come to a conclusion. Quantitative data on the other hand, can be coded in a numerical form. Quantitative methods are specific, testable outcome-oriented, involves experiments or closed questions or rating scales.

2.5 UX and Usability Techniques

With the rapid development of new technologies more and more new methods are becoming available for UX tracking and conducting usability testing. The studies on UX and Usability follows the whole product development lifecycle. Understanding actual development processes and what kind of techniques can be applied in each step is important. Usually product development follows:

- Analysis
- Design

- Implementation
- Deployment

UX and usability should also be considered in each of these steps.
Conducting user research, understanding the user goal is an essential step. It plays a decisive role in product success. Market research, collecting internal information about users, inspecting the site or the user's environment, interviewing the users are the common techniques.
Based on the analysis result, the concepts and prototypes can be designed the implementation follows naturally after. Card soring, formative usability testing, summative usability testing, customer journey mapping, heuristic evaluation, cognitive walkthrough are the common usability techniques to test and evaluate in this phase.
On deployment phase, important real-world data can be collected from users, it can assess the effectiveness of the product in the user's environment. Conducting surveys, field testing are the common techniques to address the user satisfaction and further improvement. Web Analytics is an important new type of tool to provide additional types of information such as heat maps and visitors paths.

2.5.1 Remote usability

As it is mentioned previously remote usability can be divided into moderated (synchronous) remote testing and unmoderated (asynchronous) remote testing.
Moderated remote testing doesn't differ much from the lab-based testing. The significant difference is the spatial distance. In moderated remote testing, the moderator, test participant and observers are not physically in the same space. Moderated remote testing works well with any online meeting tools, such as WebEx, Zoom, or Skype. Using one of these tools' participants can share their screen and an audio connection via meeting tool or through an external phone call, participants can also be asked to think out loud while performing the tasks. It is easy to include video of the user's face in the session as well.
Unmoderated remote testing is at some degree involves automation. Depending on the application, keystrokes, mouse clicks, taps and swipes, completion times etc., are automatically recorded together with the remote session. With the session recordings evaluators can obtain qualitative analysis. Here are some web-based companies provides similar kind of services:

- UserZoom
- UserTesting
- Hotjar

- Validately
- Userlytics

One big advantage of unmoderated remote usability testing is it is fast. Since it is conducted asynchronously multiple participants can complete their tasks at the same time. In unmoderated remote testing, not only conducting test is fast, but also the results are returned fast. It is really resource and time saving, especially in Agile development processes.

The concept of remote usability testing has broadened beyond usability testing to include user information findability, showing users clicked and how often, showing navigation paths and success/failure rates etc. With the increasingly common use of Agile development, iterative design process is also adopted to increase the usability and acceptability of the website. Understanding the user requirements in software development is very important. Taking the user requirements into consideration right from the beginning and included into whole product development cycle results in products with high acceptability after release. With conducting remote usability tests, developers and designers can address the user requirements more clearly, adjust designs and functionalities according to usability test results.

2.6 Enabling Tools for Usability Testing

According to the usability testing goals laboratories for conducting usability testing may include tools and equipment such as gaze tracking devices, communication devices, devices for measure biofeedback. As the usability assessment widely recognized as critical to the success of interactive interface design including web design. Special equipment such as Electroencephalogram (EEG), Electrocardiogram (ECG), and Electromyogram (EMG) could also be found in more advanced laboratories, they are used to evaluate individual's emotional reactions to different web interface designs (Lee and Seo 2010).

In recent years, artificial intelligence (AI) has revolutionized almost all sectors, many improvements in UX and usability testing tools and technologies included. Powerful data processing capability of AI intelligent algorithms can analyze site visitor's information, user generated data to help visitors provide more personalized and more engaging experience. Whether it is user-centered design, UX study or usability testing, the most important factor of all is the human factor. Understanding human emotional, cognitive and behavioral state of a human is critical. (Landowska 2015) studies the applicability of affect recognition methods and emotion representation models in usability testing.

Several applications scenarios are proposed concerning program usability testing and software process improvement based on multimodal emotion recognition algorithms (Kolakowska et al. 2013).
The advancement of hardware and computation speed, artificial neural network and deep learning techniques widely adopted in real world systems and become more powerful.
Today, algorithms based on deep learning can be used for face recognition to detect and identify humans, voice recognition, gesture recognition, facial expression recognition to understand sentiment, emotional state of a person. Calibration free eye tracking can also be achieved thanks to supervised deep learning.
In the following sections, all these technologies are studied in detail.

2.6.1 Artificial Neural Networks

Artificial neural networks (ANNs) or usually called neural networks (NNs) in short that inspired by the biological neural networks have been a huge success in many tech fields. Many kinds of the NN exists such as convolutional neural network (CNN) and recurrent neural networks (RNNs). Usually, CNNs are used to train models that capable of processing images with high accuracy and recurrent neural networks perform better on texts.
CNNs made up of neurons that have learnable weights and biases. Each neuron receives some inputs, performs a dot product and optionally follows it with a non-linearity. The whole network still expresses a single differentiable score function. CNNs make the explicit assumption that the inputs are images, which allows us to encode certain properties into the architecture. These then make the forward function more efficient to implement and vastly reduce the number of parameters in the network. CNNs have neurons arranged in 3 dimensions: weight, height, depth (depth refers to the third dimension of an activation function). A CNN is a sequence of layers, and every layer of a CNN transforms one volume of activations to another through a differentiable function. We use three main types of layers to build CNN architectures: Convolutional Layer, Pooling Layer, and Fully connected Layer. These layers are stacked to form a full CNN architecture.
Recurrent Neural Networks (RNNs) are popular models that have shown great promise in many Natural Language Processing (NLP) tasks. The idea behind RNNs is to make use of sequential information. RNNs are called recurrent because they perform the same task for every element of a sequence, with the output being depended on the previous computations. Another way to think about RNNs is that they have a “memory” which captures information about what has been calculated so far. In theory RNNs can make

use of information in arbitrarily long sequences, but in practice they are limited to looking back only a few steps.
RNNs have shown great success in many NLP tasks, a commonly used type of RNNs are LSTMs (Long Short-Term Memory), which are much better at capturing long-term dependencies.
The high accuracy of deep neural network comes at a cost of requiring a large amount of training data, which sometimes not available. The ability to rapidly learn from very little data is desirable for machine learning systems.

2.6.2 Convolutional Neural Network Architectures

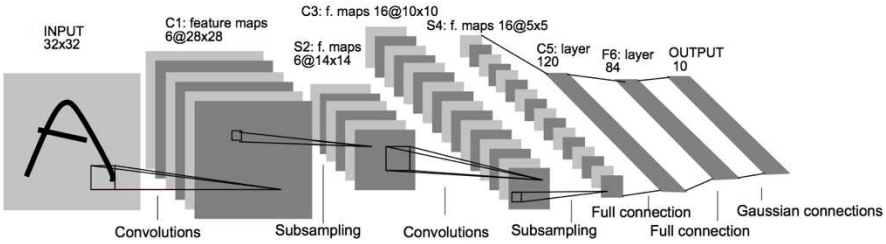


Figure 5 Architecture of LeNet-5

CNNs have wide applications in image and video recognition, recommender systems and natural language processing. The basic principles and core concepts of the CNN can be understood better from the LeNet-5 (LeCun et al. 1998). LeNet is capable of recognizing handwritten characters.
LeNet-5 comprises seven layers, not counting the input, all of which contain trainable parameters (weights). The input is 32×32 pixel image.
Over the development of CNN, researchers studied and constructed different types of CNN architectures in order to reduce the training time spent or improve the task accuracy. The most common deep learning architectures for CNN today are:

- VGG
- ResNet
- Inception
- Xception

VGG (Generosi et al. 2019) architecture is one of the first to appear, the simple architecture using only blocks composed of an incremental number of convolutional layers with 3×3 size filters. Besides, to reduce the size of the activation maps obtained, max-pooling blocks are interspersed between the convolutional ones, reducing the size of these activation maps by half. Finally, a classification block is used, consisting of two dense layers of 4096 neurons each, and the last layer, which is the output layer, of 1000 neurons.

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Figure 6 VGG Architecture

VGG has two variations VGG16 and VGG19, the numbers refer to the weighted layers that each network has, columns D and E corresponds to the networks, respectively. The input of the VGG is a fixed size 224×224 RGB image. VGG16 has 138 million parameters while VGG19 has 144 million. The rest of the architectures in the table are there because, Simonyan and Zisserman had a hard time training their architecture to converge. Since they couldn't do it, what they came up with was to train networks with simpler architectures first, and once these converged and were trained, they took advantage of their weights to initialize the next network, which was a little more complex, and so on until they got to the VGG19. It takes very long time to train this network and the number of parameters it has is very high.

The ResNet (He et al. 2015) architecture, was a milestone in introducing an exotic type of architecture based on “modules”, or as it is now known, “networks within networks”. It introduced the concept of “residual connections”. There are different variations of ResNet with different number of layers, but the most used is ResNet50, which consists of 50 layers with weights.

It is remarkable that although it has many more layers than the VGG, it needs much less memory, almost 5 times less. This is because, instead of dense layers in the classification stage, it uses a type of layer called GlobalAveragePooling, which converts the 2D activity maps of the layer in the feature extraction stage to an n-classes vector that used to calculate the probability of belonging to each class.

The Inception (Shah and Yang 2015) architecture, uses blocks with filters of different sizes that are then concatenated to extract features at different scales. The goal of the inception block is to calculate activation maps with different sized convolutions to extract features at different scales. Then the activation maps are concatenated into one activation map. It requires even less memory than the VGG and ResNet.

The network architecture Xception (Chollet 2017) is created by optimally making convolutions on Inception architecture. This is achieved by separating the 2D convolutions into 2 1D convolutions. The advantage is it takes even less time to train.

2.6.3 Face Detection

Face detection is a computer technology that determines the location and size of a human face in the digital image. The facial features are detected and any other objects like trees, buildings and bodies are ignored from the digital image. There are two types of approaches to detect facial part in the given digital image, feature based and image based (Kumar, Kaur, and Kumar 2019). Feature based approach tries to extract features of the image and match it against the knowledge of the facial features. While image-based approach tries to get the best match between training and testing images.

Active Shape Model (ASM) focus on complex non-rigid features like actual physical and higher-level appearance of features. Main aim of ASM is automatically locating landmark points that define the shape of any statistically modelled object in an image.

Point distribution model (PDM) was developed independent of computerized image analysis and developed statistical models of shape. The idea is that once one can represent shapes as vectors, after that they can apply standard statistical methods to them just like any other multivariate object.

Feature searching Viola and Jones presented an approach for object detection which minimizes computation time while achieving high detection accuracy. Viola and Jones proposed a fast and robust method for face detection which is 15 times quicker than existing techniques at the time of release with 95% accuracy. The technique relies on the use of simple Haar-like features that are evaluated quickly through the use of a new image representation.

Based on the concept of an integral image it generates a large set of features and uses the boosting algorithm AdaBoost to reduce the over complete set (Zhang, Xie, and Xu 2011). Haar-like features are digital image features used in object recognition. All human faces share some universal properties of the human face like the eye region is darker than its neighbor pixels, and the nose region is brighter than the eye region. A simple way to find out which region is lighter or darker is to sum up the pixel values of both regions and compare them. The sum of pixel values in the darker region will be smaller than the sum of pixels in the lighter region. If one side is lighter than the other, it may be an edge of an eyebrow or sometimes the middle portion may be shinier than the surrounding boxes, which can be interpreted as a nose. This can be accomplished using Haar-like features and with the help of them, the different parts of a face can be interpreted.
The detector is applied in scanning fashion and used on gray-scale images, the scanned window that is applied can also be scaled, as well as the features evaluated. This face detection framework is capable of processing images extremely rapidly while achieving high detection rates.

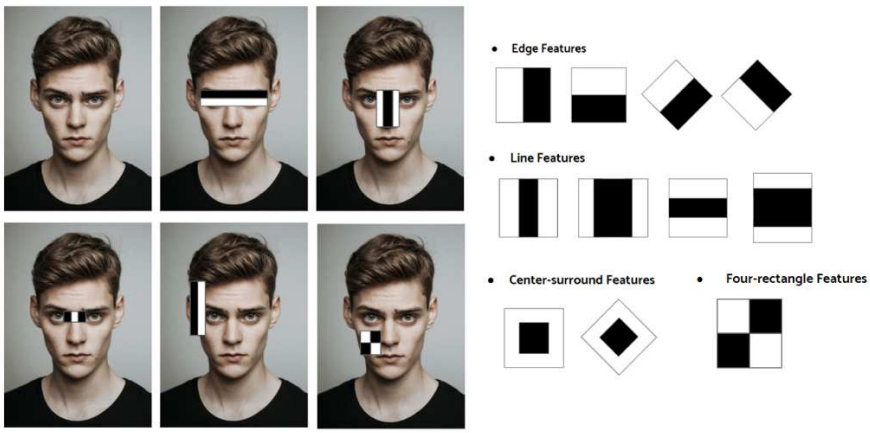


Figure 7 Haar features

Local binary pattern (LBP) technique is very effective to describe the image texture features (Ahonen, Hadid, and Pietikäinen 2004). LBPs are used to characterize the texture and pattern of an image/object in an image. They process pixels locally which leads to a more robust, powerful texture descriptor. In LBP, the face area is first divided into small regions from which LBP histograms are extracted and concatenated into a single, spatially enhanced feature histogram efficiently representing the face image. It has advantage such as high-speed computation and rotation invariance, which facilitates the broad usage in the fields of image retrieval, texture examination, face recognition, image segmentation, etc.

Histogram of Oriented Gradients (HOG) is a feature descriptor generally used for object detection. HOGs are widely known for their use in pedestrian detection. A HOG (Dalal and Triggs, n.d.) relies on the property of objects within an image to possess the distribution of intensity gradients or edge directions. Gradients are calculated within an image per block. A block is considered as a pixel grid in which gradients are constituted from the magnitude and direction of change in the intensities of the pixel within the block.

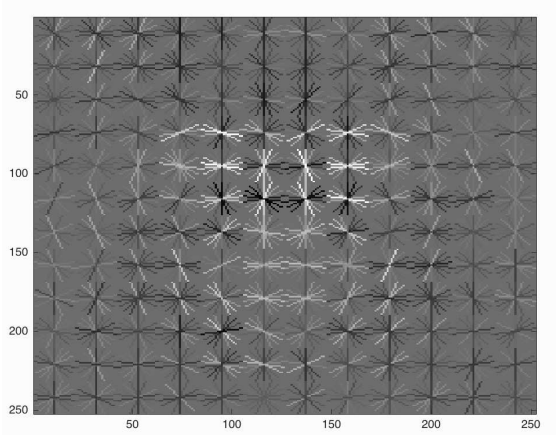


Figure 8 HOG visualization with cell size being 3

The descriptors are gradient vectors generated per pixel of the image. The gradient for each pixel consists of magnitude direction, calculated using following formula:

$$g = \sqrt{g_x^2 + g_y^2}$$

$$\theta = \arctan\frac{g_y}{g_x}$$

g_x and g_y are respectively the horizontal and vertical components of the change in the pixel intensity.

Neural Network-based Face Detection A neural-network based face detection system is introduced in (Rowley 1996), the connected neural network examines small windows of an image and decides whether each window contains a frontal human face.

It first applies a set of neural network-based filters to an image, and then arbitrates the filter outputs. The filters examine each location in the image at several scales, looking for locations that might contain a face. The arbitrator then merges detections from individual and eliminates overlapping detections. With the development NNs, Deep Learning architectures are developed and implemented in the face detection task. In DL approach templates are learned from examples in images. In general, appearance-based methods rely on techniques from statistical analysis and DL to find the relevant characteristics of face and non-face images. The learned characteristics are in the form of distribution models or discriminant functions that is consequently used for face detection. Meanwhile, dimensionality reduction

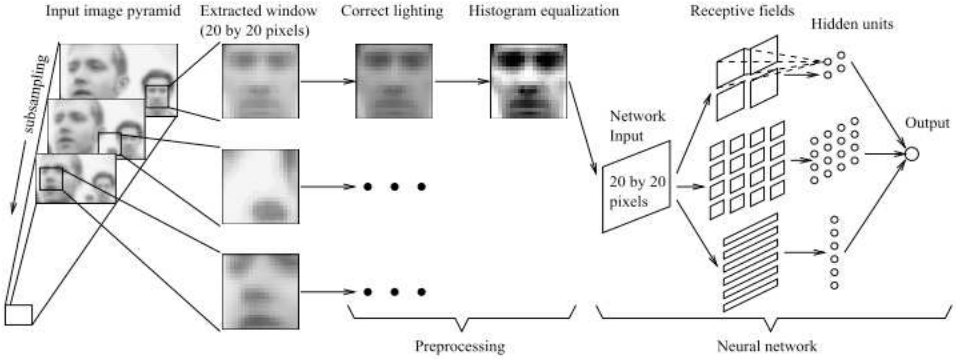


Figure 9 Neural network-based face detection

is usually carried out for the sake of computation efficiency and detection efficacy. DL methods are good at detecting small, blurred and partially occluded faces in uncontrolled environment. For example, (Detector 2018) introduced a novel context-assisted single shot face detector, named PyramidBox. PyramidAnchor, is a novel context anchor proposed in the paper to supervise face detector to learn features from contextual parts around faces. It has over 95% of detection accuracy.

2.6.4 Age and Gender Detection

Automatic age and gender classification has become relevant to an increasing number of applications, particularly since the rise of social platforms and social media. All humans have two different of ages – chronological and biological. Chronological age is the number of years a person has been alive, while the biological age refers to how old a person seems. The age detection in this context refers to the biological age. Demographics is important when conducting usability tests. Clustering analysis can be conducted on top of behavior analysis to see if there is some kind of correlations exists. Both age and gender of a person can be classified by learning the facial representations from the images through the use of deep CNNs. A significant increase in performance can be obtained on these tasks with the help of DL. Deep residual regressors and DEX pipeline approaches are compared in (Agustsson et al. 2017) and reported that the best model DEX Mean Absolute Error is 4.082 on IMDB-WiKi dataset. For the gender detection, (Dehghan et al. 2017) reports their model on gender detection on Adience benchmark resulted 91%.

2.6.5 Facial Expression Recognition

Usability tests are usually conducted and managed by usability experts who typically had education and training as cognitive scientists, experimental psychologists, or human factors engineers. It is important to observe the facial expression of the participant in order to understand their cognitive states during the test.

The facial expression recognition refers to the process of recognizing and classifying the human facial expressions from images or video streams. American psychologist and professor emeritus at the University of California, Paul Ekman is a pioneer in the study of facial expressions and their relation to emotions. He studied Universals and cultural differences in facial expressions of emotions, in his study he concluded that the facial expressions are a universal system of signals which reflect the moment-to-moment fluctuations in a person's emotional state. He long theorized a discrete set of physiologically distinct emotions: anger, disgust, fear, happiness, sadness and surprise. The facial expressions detected can be classified into one of the corresponding emotion categories. The Ekman's set of emotions is surely one of the most considered ones.

Nowadays, understanding emotions often translates into the possibility of enhancing human-computer interactions (HCI). A variety of technologies exists to automatically recognize human emotions, spanning across facial expression analysis, acoustic speech processing and biological responses interpretation.

Besides suggesting the primary set of six emotions, Ekman also proposed the Facial Action Coding System (FACS), which puts facial muscles movement in relation with a number of Action Units (AU) areas; each facial action unit identifying an independent motion of the face. Movements in facial muscles are perceived as changes in the position of the eyes, nose, and mouth. By capturing images of the user's facial expressions and head movements, the system can detect the corresponding action units of the eyes, mouth, and nose, change in the position of the dots that represents the action unit in a coordinate system. The changes detected then can be analyzed and interpreted (Gonzalez-sanchez et al. 2017).
The work of Paul Ekman is still considered to be the core of Emotion recognition based on the facial expressions.

Figure 10 Ekman's Universal Facial Expressions (from top left Anger, Fear, Disgust, Surprise, Happiness, Sadness)

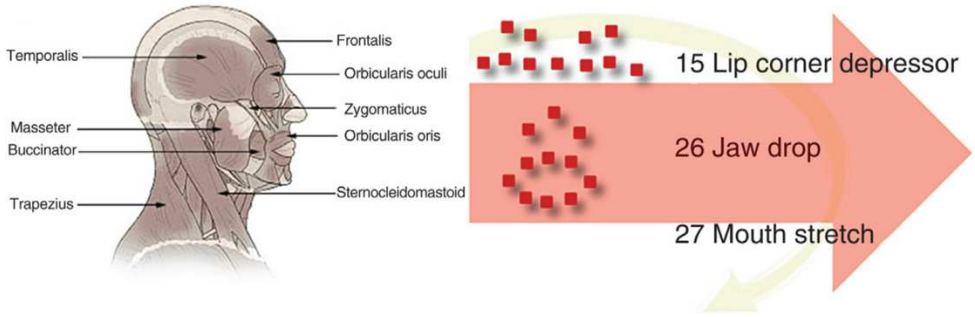


Figure 11 Movements of individual facial muscles are encoded as action units

Another alternative to the Ekman's discrete emotions or categorical model is proposed in the article (Mollahosseini, Hasani, and Mahoor 2017). The model predicts the intensity of valence and arousal in addition to the Ekman's categorical classification.

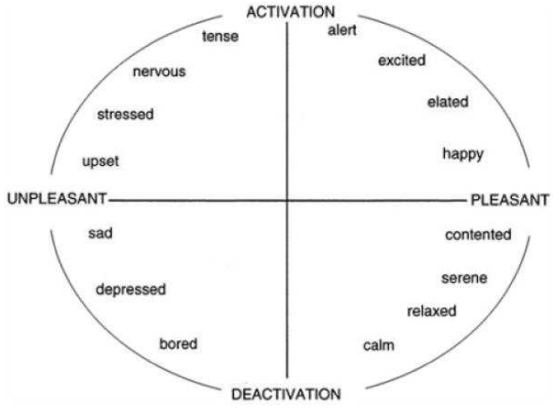


Figure 12 The circumplex model of Russell

Valence refers to how positive or negative an event is, and arousal reflects whether an event is exciting/agitating or calm/soothing (Russell 1980).

Emotion valence simply differentiates between pleasant (positive) and unpleasant (negative) feelings. To have a view on the valence dimension is often useful, as non-basic affective states may appear, not forgetting that even basic emotions frequently blend together.

User engagement tells how much the user is captivated by an experience, and thus reveals to be another valuable tool in marketing contexts. Engagement cannot be uniquely

defined; hence various methodologies can be applied to estimate how much the user is attracted by a product. In this context, it is considered that engagement mostly related to the visual contact of the user with what is in front of him/her. According to this, engagement can be measured by processing video frames and combining information about gaze and emotional status.

Thanks to recent, continuous improvements on the ImageNet challenge (Krizhevsky, Sutskever, and Hinton 2012), application of Deep Learning (DL) algorithms has emerged as a trend among the appearance-based ones.

A lot of DL architectures have been proposed and employed for facial emotion recognition, each of which was outperforming its ancestors, thereby constantly improving state-of-the-art accuracies and performances.

Up until now, the most efficient DL architectures have been observed to be Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) networks. Hybrid frameworks do exist too, in which different architectures characteristics are combined together in order to achieve better results.

State-of-the-art CNNs have reached a noteworthy classifying capability, performing very well in controlled scenarios. Research has hence shifted toward the categorization of emotions in the wild with an accuracy possibly high as much.

In the wild-related issues arise owing to subject pose, environment illumination and camera resolution, to compensate for which it is necessary to train networks on very large and condition variability rich datasets. On the other hand, training images need to be properly labeled by expert FACS coders; therefore, to extend a facial expression dataset is a complex and costly procedure. That is why emotion classification in not controlled scenarios still remains a challenging terrain for computer vision research.

2.6.6 Gaze Tracking

Gaze tracking has been studied for decades in Human Computer Interaction (HCI) and Computer vision field. Gaze tracking technology consists of various components that work together to track where people's visual attention is concentrated. These components are cameras or infrared-based eye trackers. Advancement in image processing algorithms based on DL can also be used to interpret the gaze points, fixation sequences, and areas of interest from the camera captured data. Most of the gaze tracking tools starts gaze tracking with a calibration phase. They detect the face and eye regions of the user from the camera stream, then maps the gaze coordinates on a screen.

A common methodology called Pupil Center Corneal Reflection (PCCR) involves using NIR LEDs to produce glints on the eye cornea surface and then capturing images/videos

of the eye region. Gaze is estimated from the relative movement between the pupil center and glint positions. The gaze location of a user depends both on the gaze direction and also on the head orientation. In PCCR techniques, if the user moves their head with respect to the tracker-camera axis while looking at the same point on the screen, the glint vectors with respect to the pupil centers will be different from each other. Therefore, the estimated gaze locations will be inaccurate. 2D regression, 3D model, and Cross ratio based methods all belongs to the PCCR based methods that use NIR illumination to estimate the gaze direction or the point of gaze using polynomial functions, or a geometrical model of the human. In 2D regression based methods, the vector between pupil center and corneal glint is mapped to corresponding gaze coordinates on the frontal screen using a polynomial transformation function. They utilize the features of the human eye, like eye geometry, pupil contours and corneal reflections and can be implemented using single camera and a few NIR LEDs. However, it is very vulnerable to head movements and requires users to hold their head very still using a head rest, chin rest or bite bar. 3D model based methods use a geometrical model of the human eye to estimate the center of the cornea, optical and visual axes of the eye and estimate the gaze coordinates as points of intersection where the visual axes meets the scene. They have tolerance towards user head movement but the hardware requirements for implementing them are high. It also needs several light sources or multiple cameras. Cross ratio based methods work by projecting a known rectangular pattern of NIR lights on the eye of the user and estimating the gaze position using invariant property of projective geometry. They do not need an eye model or hardware calibration and allow free head motion. But they are affected by problems such as increased error with distance of user and user dependent factors eye (Kar and Corcoran 2017).

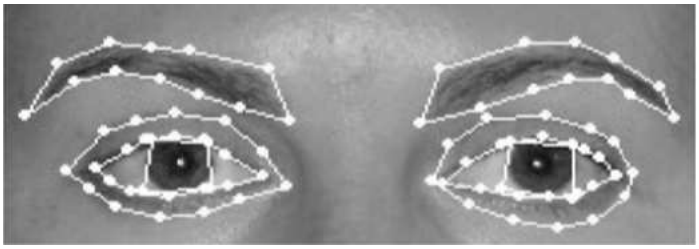


Figure 13 Image fitted with an Active Appearance model of the eye region

Another gaze tracking approach commonly known as appearance-based approaches in which the information from the eye region is represented using a model trained with a set of features extracted from the eye images.

Deep learning techniques with CNNs have been implemented in estimation of gaze. CNN models are used to learn the mapping from eye images to gaze position from the images taken with low-cost webcams. (Krafka, Khosla, and Kellnhofer, n.d.) published a large-scale gaze tracking dataset and eye tracking CNN called iTracker. The dataset contains 1450 people consisting of almost 2.5 million frames. The iTracker achieves a significant reduction in error over previous approaches while running in real time (10-15 fps) on a modern mobile device. The prediction error iTracker produced is 1.71cm and 2.53cm without calibration on mobile phones and tablets respectively. With calibration, the error can be reduced to 1.34cm and 2.12cm. the features learned by iTracker generalize well to other datasets, achieving state-of-the-art results.
In gaze tracking literature, the gaze tracking accuracy measures are presented in different ways such as angular accuracy in degrees, distance accuracy in cm or distances in pixels. But is most typical eye gaze tracking validation operation, a user gazes at an interface on a computer screen which provides him/her with visual stimulus in the form of a set of targets. Gaze tracking accuracy is estimated as the average difference between the real stimuli positions and the measured or predicted gaze positions, which also provides an idea about the performance of the system.

2.7 Implementation of Deep Learning in Usability Testing

Over the past years producers and the marketing/branding industry demonstrated an increasing interest in emotional aspects of user behavior, as understanding the emotional state of users is crucial for developing successful products and services (Ganglbauer et al. 2011). Psychophysiological methods may offer data throughout the process of experience, which unfolds new possibilities for improving remote usability testing and UX. This motivated the demand for automatic emotion recognition techniques as a tool for getting a larger quantity of more objective data. The attempt to implement face detection, facial expression recognition and gaze tracking techniques to explore the user insights, capture user interactions and use these data to enhance the modern-day remote usability testing is an important innovation in usability testing.
The importance of emotions in usability testing in well known for a long time. From the time that usability testing was born, it is important to monitor the test participants facial expressions along with their behaviors to assess the usability testing results. That's why the usability experts are chosen from who had education and training as cognitive scientists, experimental psychologist, or human factors engineers. Facial expression recognition and gaze tracking technologies can help to deliver a greater level of insight into behavior patterns and also can reflect psychological and emotional state of the user.

Several studies, such as (Georges et al. 2016) and (da Silva Franco et al. 2019), proposed systems that allow us to correlate data from different typologies, e.g. eye-tracking fixations, sentiment analysis, body gestures, or facial expressions. However, these systems are designed to support only formal UX evaluation assessment and not for conducting usability testing. The integrated use of gaze tracking and emotion recognition systems based on deep learning algorithms, to support the collection of relevant information useful for remote usability testing on both laptop and mobile devices.

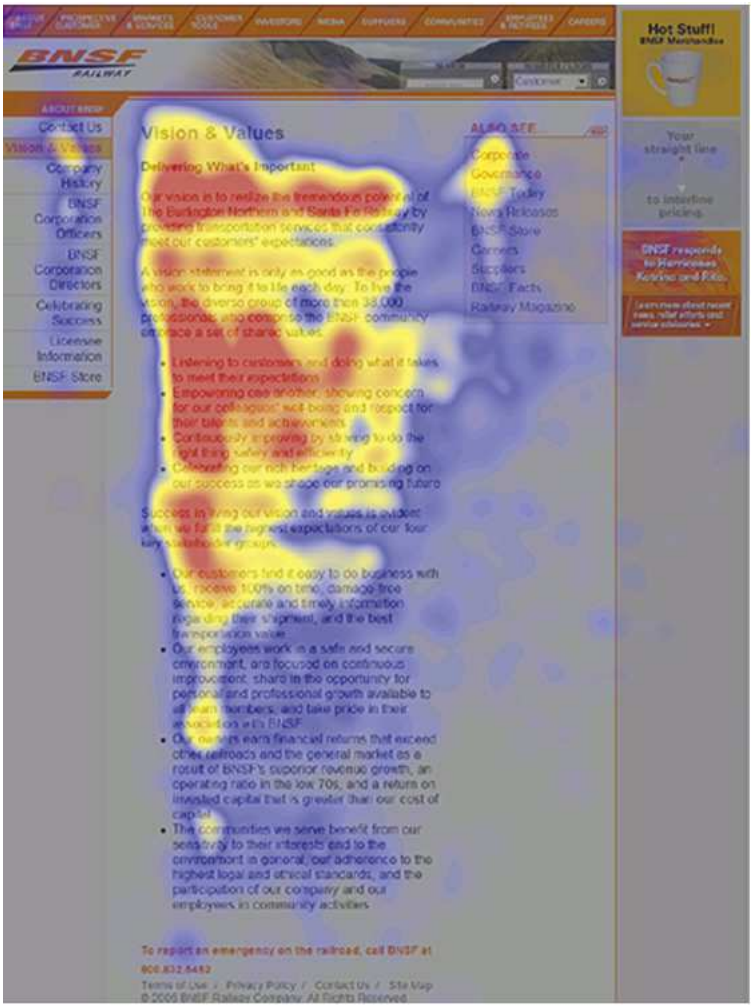


Figure 14 A heatmap shows eye fixations on a website

As it is stated previously, remote usability testing should address effectiveness, efficiency and satisfaction aspects of the testing target. Gaze tracking is a powerful and very useful technology has been implemented by the researchers when conducting usability tests. However, all the gaze tracking in the studies is conducted by special equipment like Tobii Pro or eye-tracking glasses. This type of equipment allows test conductors to see where users see and where they look by tracking their eye movements and the length of time they fixate on a certain part of the screen. A color-coded heatmap of the screen shows the hotspots – the areas of the screen that the longest time for fixations – for individual users and combined for all users in a study. Calibration is required before using this type of equipment and the user can move naturally without breaking the calibration.

The red color on the heatmap shows the highest concentration for fixations, followed by yellow and then purple. It reveals that users focused their attention on the information at the top and partway down the left side of the page (CAROL 2020).

Another output from eye tracking is a gaze plot, which shows the order in which users moved around the website (CAROL 2020). The gaze plot example is shown on the following. It shows the users review on the website. Different colors represent the different users, the size of the circle represents how long a user looked at the spot and the numbers represent the order in which a user moved around the website.

Costs to purchase the gaze tracking technology are still high. However rental options have made the price and access to the equipment more affordable.

AI based gaze tracking is the most cost-effective option compared to other methods. With the advancement of deep learning and high accurate gaze tracking datasets, calibration free webcam-based gaze tracking models are also can be trained and used in usability testing.

An important usability matrix studied is satisfaction, in most cases it is measured by asking test participants to answer a questionnaire. It is an affordable way of getting the satisfaction result, but several disadvantages exist. Dishonest answers, unanswered questions, differences in understanding and interpretation, hard to convey feeling and emotions. Furthermore, some questions are difficult to analyze, there is no way to know if the respondent has really understood the questions. Emotion recognition, sentiment analysis technologies have already been implemented in tracking customer satisfaction in various context. By implementing similar technology in remote usability testing, usability satisfaction matrix can also be measured.

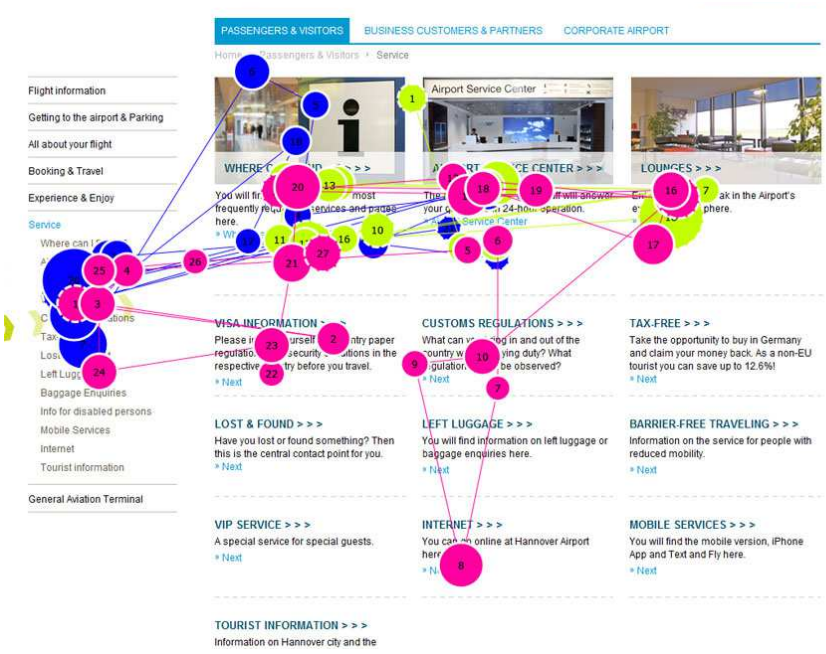


Figure 15 A gaze plot shows the order of the users gaze and relative length of time at each point

Ekman's universal facial expressions model and circumplex model of Affect are two important models that can be used to extract peoples emotional state by monitoring their facial expressions. and their facial expressions to their emotional states.

Usability testing on mobile devices typically requires a few adaptations from standard desktop testing. Specialized equipment is helpful to hold the device in place or keep the device within the range of a camera so that you can capture everything in focus. There are software tools that provides mirroring a mobile device on computer and record it, then the recordings are analyzed manually to conclude the test result. However, this method requires resources for analyzing the recordings and it is time consuming. Because testing mobile devices is rapidly expanding growth area in UX research, a variety of inexpensive technologies have become available to make it relatively easy to set up and use.

Mr.Tappy and MOD 1000 are two types of tools used in mobile testing. The former has a document camera to capture user interactions and user must perform all the tasks under the camera, while the later has a base station where the mobile device is placed stationary under a camera, but the user is allowed to move freely.

It is also possible to capture user interactions on mobile devices with the help of programs developed from scratch. A mobile SDK is introduced in the 4th chapter that has the capability of capturing user interactions together with their facial expressions and gaze tracking information (eye fixation).

Chapter 3.
Preliminary AI-based Technology Development and Testing

The training process of deep learning models will be implemented in the final Usability Testing Platform is explained in detail in this chapter. How the datasets are constructed, how the facial expression recognition, age and gender recognition and gaze tracking models are trained with supervised learning, how the preliminary tests using the models are conducted is described.

To better understand and familiarize with the modern usability testing processes, to explore the integration possibilities of trained facial expression recognition, age and gender recognition, gaze tracking models, a preliminary test platform is developed. A demo experiment is conducted. An online home electronics store is set up on the Amazon Server. A small team of participants are asked to buy a washing machine from the online store and all their behavior on the online store is monitored. The data captured during the test is analyzed to validate the prototype and usability of the online store.

3.1 Deep Learning Models

3.1.1 Facial Expression Recognition Model

All the facial expression recognition models have reached different accuracies according to the datasets they have trained on. It has been observed that all the models with high accuracies are trained on the lab generated datasets such as MMI, CK+. However, the models trained with the datasets within the wild properties (usually web crawled face images) have lower accuracies, that's because most of the datasets collected from the world wild web have inaccurate labels (Barsoum et al. 2016). Moreover, the level of exposure of a human face on an image could also lead to inaccurate recognitions.

Experiments are conducted with a dataset constructed by merging the lab generated CK+, the re- tagged FER and AffectNet. The assumption is that by combining the lab generated highly accurate dataset with the "in the wild datasets", it may result a better accuracy model for the in the wild benchmarks.

Datasets play a crucial role in supervised learning; the neural network models depend greatly on them. There are many public datasets for facial expression recognition, since

most of them are prepared by web crawled face images with emotion related keywords, the label accuracy is not very high. The lab generated datasets like CK+ (Lucey et al. 2010) on the other hand, has a high label accuracy but the dataset size is small. FER+ dataset is the re- tagged version of original FER dataset with crowd sourcing. It has a label accuracy over 90% but it contains only about 35k images (Barsoum et al. 2016). AffectNet (Mollahosseini, Hasani, and Mahoor 2017) dataset has over 1 million web crawled face images, it also contains 450k categorically annotated images by expert human labelers. For our study we have examined all the images and implemented a script to discard all the photos without faces or with multiple faces.

A new dataset is constructed by merging filtered AffectNet, CK+ and FER+ images tagged with one of the happy, surprise, sad, anger, disgust, fear and neutral tags. The dataset includes over 260k images.

Facial alignment techniques are used during the construction of the dataset to improve the accuracy of the dataset during construction. Given a set of facial landmarks, the facial aligner wraps and transforms the image to an output coordinate space. All faces across entire dataset are centered in the image. The images are rotated so that eyes lie on a horizontal line and the facial images are scaled to 64x64 pixels such that the size of the faces are approximately identical.

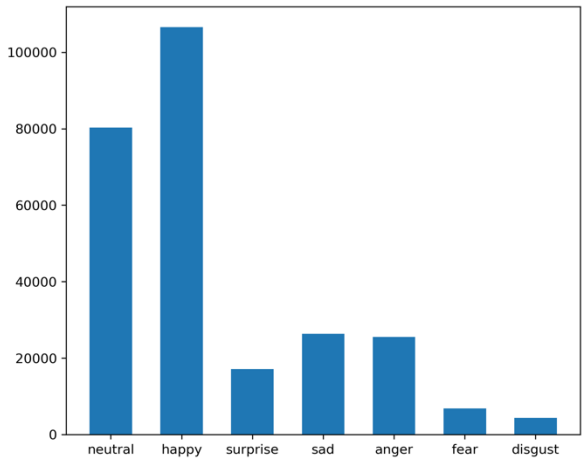


Figure 16 Distribution of dataset

Network hyperparameters are initialized as it is stated on (Barsoum et al. 2016), then the other variations are also experimented such as validation split 0.1, 0.2, number of epochs 30, 50 and 100 and dynamic learning rate defined as

$$lr = lr \times (1 - \frac{\text{epoch}}{\max epoch})$$

Learning rate *lr* is initialized with 0.025 and updated on each epoch accordingly.
State of the art deep learning models perform well on image recognition should also perform well in facial expression recognition task; that is basically the visual discrimination of human emotions is an image classification task.
VGG architecture with different depth configurations, version 2 and version 3 variants of Inception architecture are experimented on the dataset constructed.

Architectures	Accuracy (%)
VGG13	**75.48**
VGG16	74.48
VGG19	73.14
InceptionV2	75.26
InceptionV3	67.20

Table 1 Performance of different models

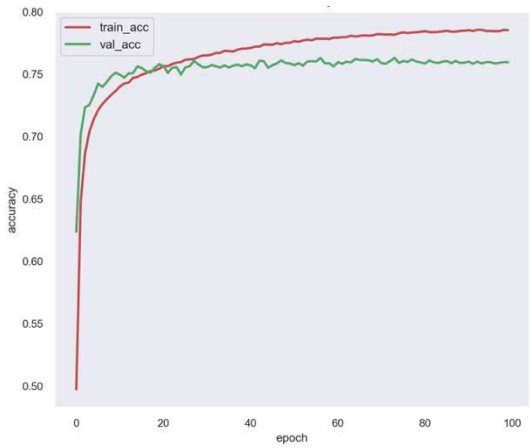


Figure 17 Training and validation accuracy of VGG13 for facial expression recognition

The table reports neural network architectures that are used to train facial expression recognition models. Different types of models are trained using the dataset. However, their performance on facial expression recognition task is different. Among them, VGG and InceptionV2 resulted in better testing accuracies.
The best performance gain is achieved by the VGG13 architecture. The training and validation accuracy of the best model VGG13 is plotted as a reference to the conducted experiment.

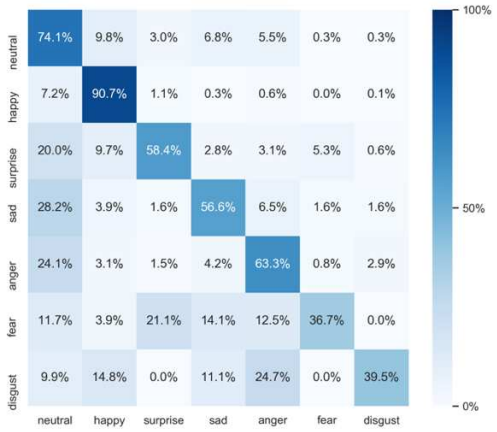


Figure 18 Accuracy of each emotion category

Confusion matrix of the VGG13 classification result is plotted as heatmap in the following. The classification accuracy of fear and disgust categories are low. The images with fear tag misclassified as surprise and disgust tag misclassified as anger has over 20% rate.
The model VGG13 we trained is also evaluated on EmotioNet (Benitez-Quiroz et al. 2017) 2018 challenge dataset: the Ohio State University, on their website, has in fact made available their dataset to give anyone the opportunity to compare their results with those of the challenges of 2017 and 2018. The table shows the evaluation result.

Categories	Accuracy	F1
happy	**0.9770**	**0.9799**
anger	0.75	0.1198
disgust	0.0128	0.0099
sad	0.5955	0.2888
surprise	0.7059	0.3944

Table 2 Evaluation results

3.1.2 Age and Gender Recognition Model

Age and gender estimation is performed by a CNN trained from scratch based on the IMDB-WIKI dataset adopting the Wide Residual Network (WideResNet) architecture and adding two classification layers: one with 101 outputs for age estimation from age 0 to 100 and another with 2 outputs for gender classification male and female. The depth and width factors of the network are set to 16 and 8, respectively. The model is trained with face images aligned and scaled to 256x256 pixels, and outputs probabilities in percentages of age and gender.

3.1.3 Gaze Tracking Model

A program is developed solely for the purpose of creating a new dataset for training the gaze tracking model. Volunteers are asked to look at a specific small circle randomly appears on the screen and taken photos of them while they are looking at the spot. For each volunteer, the small circle appears on 30 different screen positions. The position coordinates of the small circle appear on the screen are stored first and the photos taken are labeled with the position coordinates.

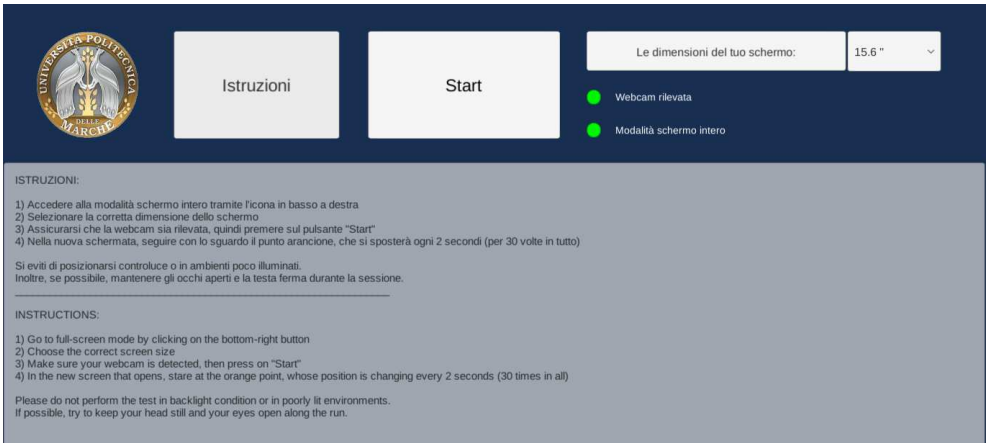


Figure 19 Web application used to collect gaze tracking dataset images

When the screen sizes are concerned, the aim of the application is trying to cover different types of screen sizes and collect more generalized data. However, due to technical limitations caused by the use of client-side languages such as JavaScript to retrieve the necessary data, it is not possible to retrieve the physical dimensions of a screen from a web application. For this reason, the application asks the volunteer to choose their corresponding screen size before starting the procedure. Knowing the

physical dimensions of the screen and the display resolution (which instead can be determined automatically), it is possible to indicate the coordinates of the points in pixels or in cm relative to the screen top left corner.

The 2D Euclidean distance function can be used as error function for the application.

$$Error = \sqrt{(x_{pred} - \hat{x})^2 + (y_{pred} - \hat{y})^2}$$

x_{pred} and y_{pred} are the predicted coordinates of the small circle, $\hat{x}$ and $\hat{y}$ are the actual coordinates displayed on the screen.
Gaze tracking model is trained adopting the architecture similar to the one proposed in (Krafka, Khosla, and Kellnhofer, n.d.) and it is based on AlexNet (Krizhevsky, Sutskever, and Hinton 2012).
On the overview of the gaze tracking CNN, CONV represents convolutional layers (with filter size/number of kernels: CONV-E1, CONV-F1: 11 X 11/96, CONV-E2, CONV-F2: 5 X 5/256, CONV-E3, CONV-F3: 3 X 3/384, CONV-E4, CONV-F4: 1 X 1/64), while FC represents fully connected layers (with sizes: FC-E1: 128, FC-F1: 128, FC-F2: 64, FC-FG1: 256, FC-FG2: 128, FC1: 128, FC2: 2). A dropout layer was added too right after each last convolutional layer. Face grid vector elements represent, in pixels: *xf*: *x* coordinate of top left point of the face box, *yf*: *y* coordinate of top left point of the face box, *w*: width of the face box, *h*: height of the face box. The output is the distance, in centimeters, from the camera.

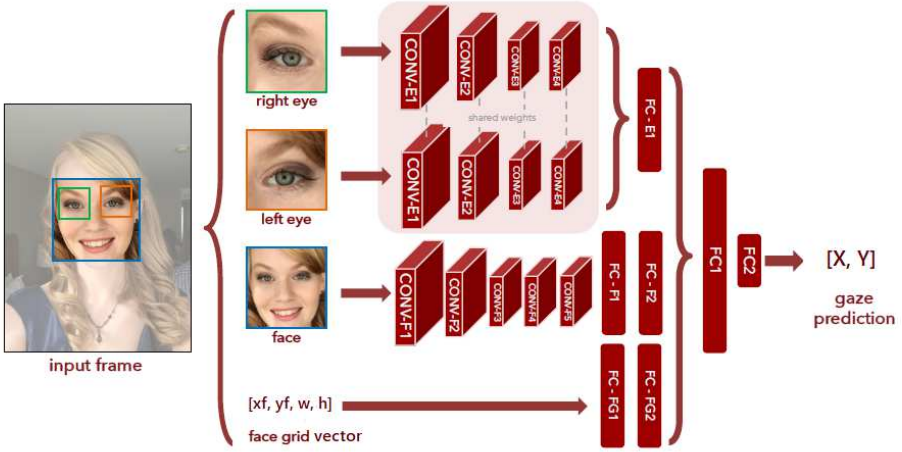


Figure 20 Overview of the gaze tracking CNN

After completing the training of gaze tracking model, a study is conducted to evaluate the tracking performance. Same software implemented to gather training samples is used in the performance evaluation. Eyes are detected using Dlib landmarks, and then cropped with the developed Python script, then fed to the model to make the prediction of the *x,y* coordinates.
The results are reported in centimeters, top left corner of the screen is considered as the origin. A total of 20 subjects agreed to take part in the test and for each participant the predictions have been made for 30 different screen positions.
To calculate the error between the predicted point and the real one following formula is used:

$$e_m = \frac{\sum_{i=1}^{n} |x_{i\,real} - x_{i\,pred}|}{n}$$

Where $x_{i\,real}$ is the real coordinate where the subject i is looking at, $x_{i\,pred}$ is the predicted coordinate, n represents the total number of participants, and e_m is the mean error (in centimeters) that is calculated for any screen position. The same formula is used to evaluate the error on y coordinate.

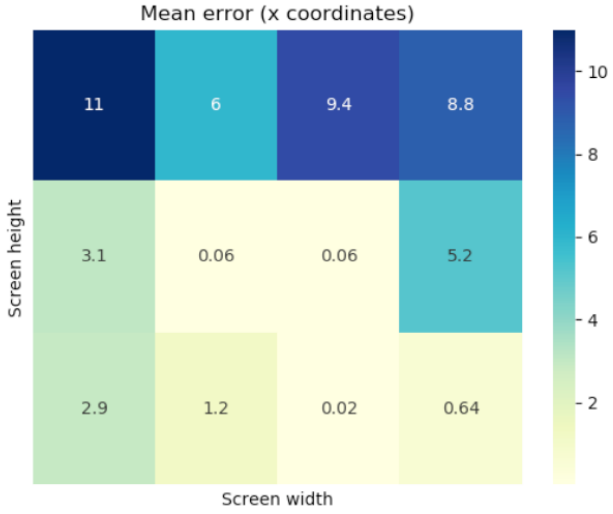


Figure 21 Mean error (cm) for each screen area for x coordinate

After calculating all the errors for participants, the mean errors have been aggregated in 12 more meaningful values, as no significant variations have been found in comparison with the high number of screen positions. Each of them represents the error in a specific screen area.

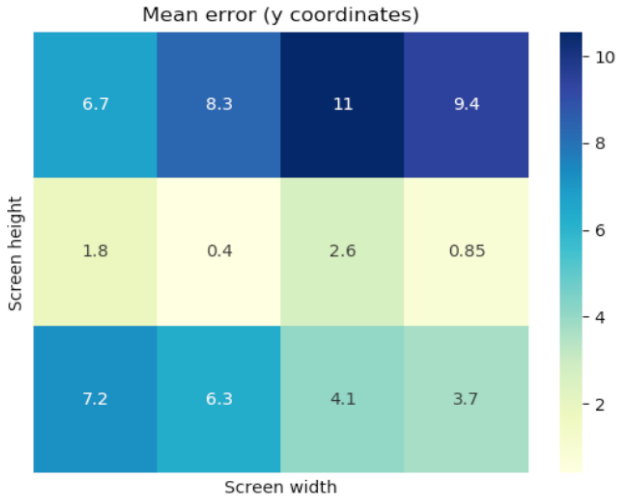


Figure 22 Mean error (cm) for each screen area for y coordinate

The heatmaps in higher error value in the top side of the screen (up to 11cm), but interesting results are obtained for the other screen positions. The error ranges between 0.02cm to 7.2cm on some areas. As the heatmap indicates, the model still needs some more improvements since the accuracy strongly depends on the area the subject stares at, but this experiment proves that it is possible to reach very accurate gaze tracking results.

3.2 Preliminary Tests

3.2.1 Testing Scenario on PC and Results

Tobii gaze tracking device is used during the test to reveal the participants' interaction path and gaze tracks during the test with the online store better. The tests are conducted on both laptop and mobile devices. There are two test scenarios in total, one for each device. The test procedure is defined as:

Task 1 → Search for a washing machine

Task 2 → Confirm the washing machine to buy and select it

Task 3 → Buy the washing machine selected on the previous step

Online home electronics store is online on www.onlinestore.it

Task 1 specifications:

Participants should search for a washing machine of A+++ class with the price not more than 400€.

During the test facial expressions of the participants are also recorded and with the help of trained CNN models, emotions are extracted, and corresponding graphs are generated. Based on the extracted emotions Valence graphs is also drawn to visualize the positivity or negativity of the participants' experience.

The most important part in the experiment conducted is to understand whether the gaze heatmaps and gaze tracks captured can be related with the emotions to reveal more hidden, and difficult to find problems.

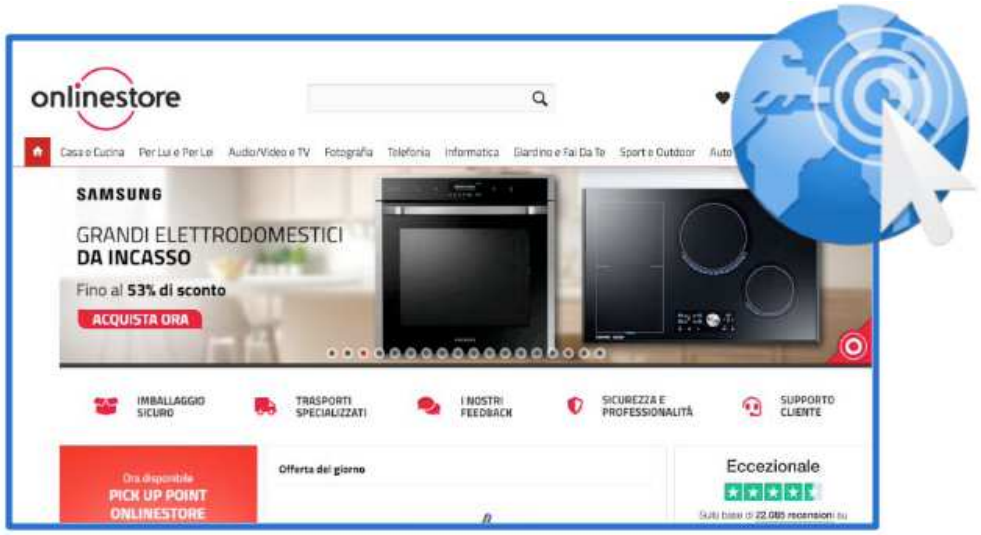


Figure 23 Online Store

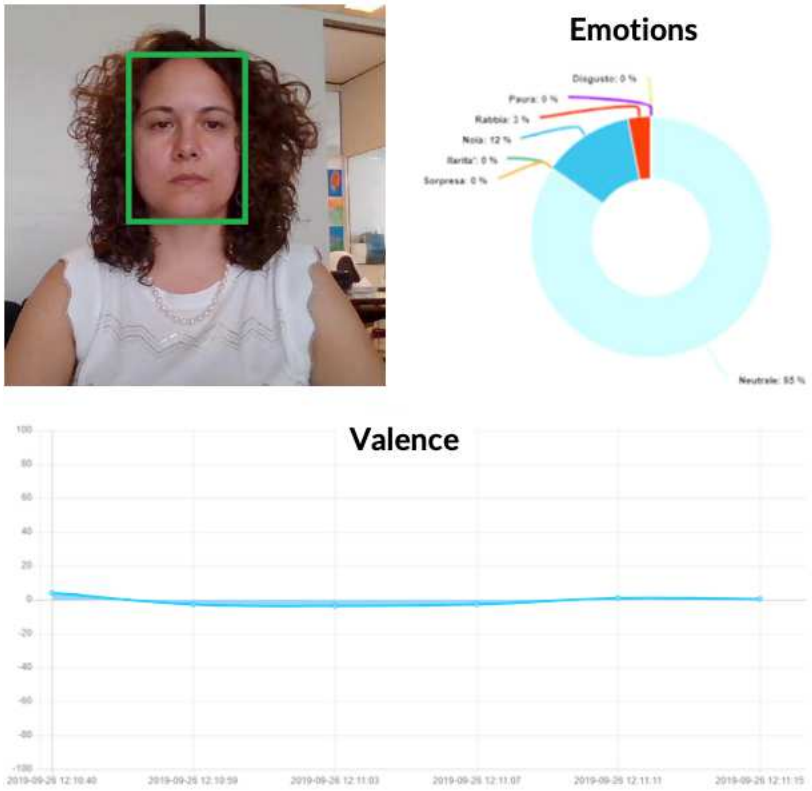


Figure 24 Emotions extracted together with the valence graph for Task 1 on Desktop

It can be observed from the valence graph that during completion of the first task, the participant's detected emotions are mostly neutral. As a result, from the Emotions graph it can also be seen that the Neutral takes up 85% of the whole graph.

The gaze path and corresponding gaze heatmaps provide more information on the task. Gaze path is the track of the gaze from the moment participant started looking at the page until he or she leaves. All the points associated with a timestamp and connected in timely order. Intensity of the colors on the heatmap indicates how long the participant is stared at the area, the longer the time is, the color becomes more red.

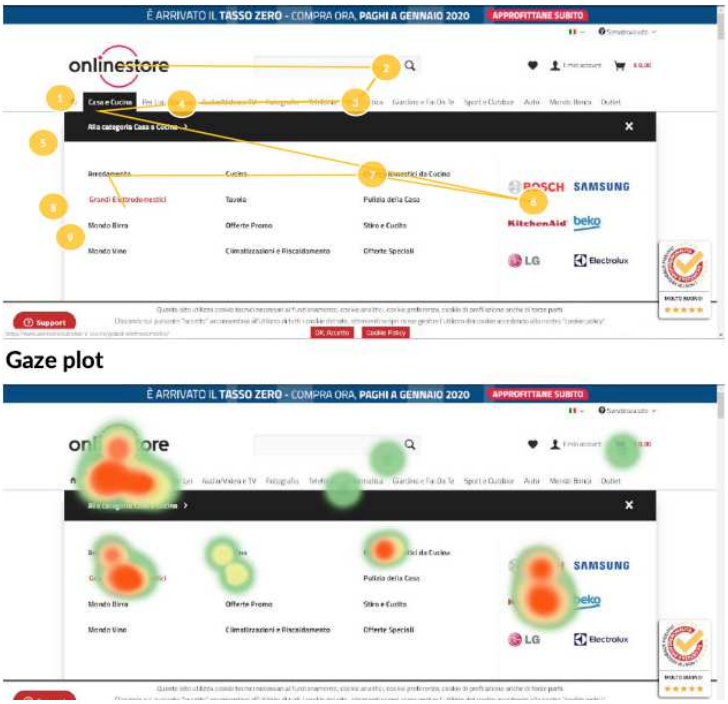


Figure 25 Gaze plots of Task 1

Figure 26 Task 2 results

Data captured during the task 2 indicates there is not much significant change in the status of the participant.

Figure 27 Task 3 result

From the task 3 plots, it can be observed that the participant expressed some negativity during the completion of the task. 16% of anger is detected from the participants facial expression, the Valence graph also shows relatively negative results compared to the previous two tasks.
The following table concludes the test results and observed parameters.

Test Results	
Number of Participants	5
Gender	2M-3F
Average age	31
Average task execution time	56s
Expert user task execution time	14s
Effectiveness	25%
Average valence	1%
Average engagement	24%

3.2.2 Testing Scenario on Mobile

The only deference on this test from the previous one is the device used. The tasks defined are the same for Mobile (Android and iOS). The same types of data are collected during the test and results obtained are similar to the previous scenario. Facial expressions and gaze tracks of the participant are monitored and analyzed. The Valence graph is also drawn to visualize the positivity or negativity of the participants' experience on Mobile device.

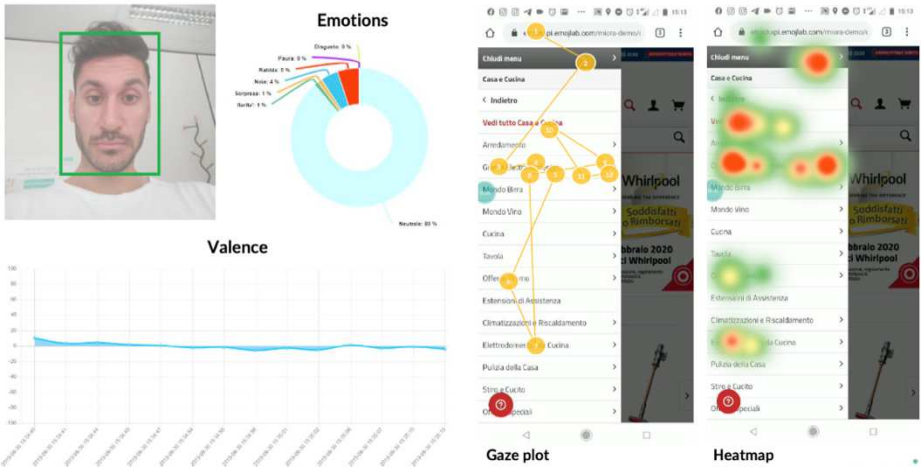


Figure 29 Task 1 results

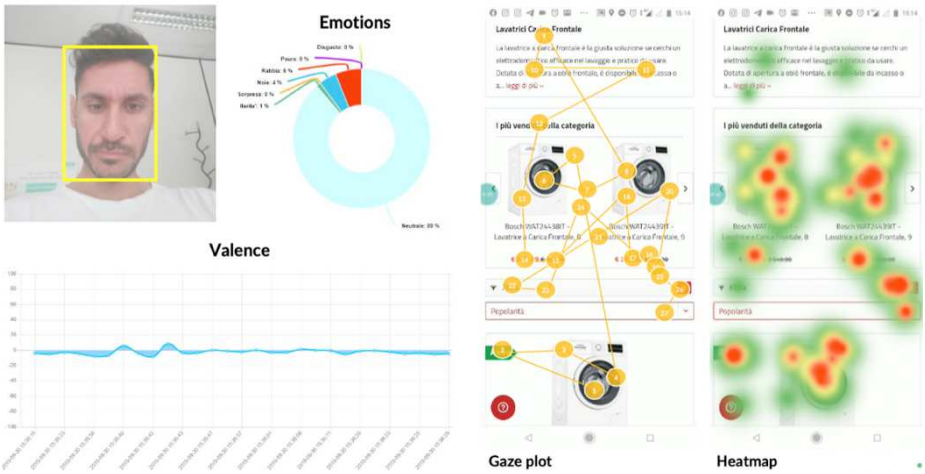


Figure 28 Task 2 results

Task 2 gaze plots shows the participant had some difficulties on completing the task. It seems he struggled a little bit on selecting the washing machine that meets the requirement defined on task 1 specification. Task 3 result shows even more negativity were present at the end of the task that frustrated the participant.

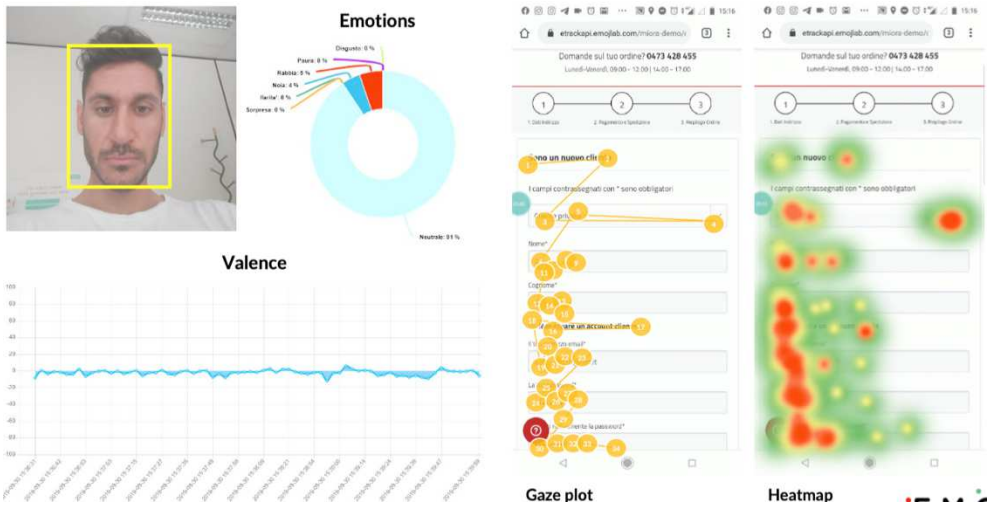


Figure 30 Task 3 results

Final result of the test scenario on mobile is concluded on the following table.

Test Results	
Number of Participants	5
Gender	3M-2F
Average age	30
Average task execution time	58s
Expert user task execution time	20s
Effectiveness	34%
Average valence	2%
Average engagement	18%

3.3 Towards an Integrated Platform

The usability test experiments conducted demonstrates the effectiveness of the approach taken. More effective integration of different modules and their organization in the whole system is predicted. The system can be improved further in terms of performance. A

dashboard with more features and appealing graphics can be developed. This is necessary to convey the test results better.

Relatively reliable results can be obtained by age and gender detection. However, gaze tracking results still varies from the results obtained by Tobii gaze tracking device. There are still have space for improving the gaze tracking model.

With the help of deep learning models more information on user behavior can be obtained; it is particularly important for the test moderator since he/she needs to understand why the participant is behaved in that manner.

The proposed enhancements in remote usability testing are effective in preliminary tests. Specific features of the new platform definition, database design, architectural structure and development procedures are described in detail in the following chapter.

Chapter 4. Platform Design and Architecture

The remote usability testing platform is developed based on artificial intelligence; specifically, deep learning approach is taken to develop the platform. It is a proof-of-concept platform, with which preliminary qualitative assessments are obtained. It demonstrates the feasibility of DL and remote usability tests can be integrated. Computer vision and state of the art image processing techniques are implemented. The platform is able to evaluate the test results automatically and visualizes them on the dashboard. In traditional usability testing approach, it requires a trained usability testing expert who had education and training as cognitive scientists, experimental psychologists, or human factors engineers to conduct mange and evaluate the test results which usually takes at least couple of hours or even days. An advantage of this usability testing platform is that everybody can conduct the usability tests, all the results are generated with the help of pre-validated deep learning models trained in the related context, results are generated in real time. It makes it easier for both test participants and test conductors.

4.1 Platform Features

With the usability measurements and UX insights that are defined in the previous chapter in mind, the data needed to generate good usability test results and their collecting methods are defined.

The Usability measures should address the effectiveness, efficiency and satisfaction aspects of the test target. The measures such as task success rate, task duration, error rate and satisfaction rate are specifically defined for the usability purpose. The Insights and Engagements are to help visualize the participants gaze tracks and interaction paths. They provide valuable information and feedbacks on the structuring of the testing target and how the users interact with it. Following the definition of the data collection methods and obtainable measures separately, some other important measures could reflect more than one aspect of the metrics are also defined in the additional list. These additional measurements are also play important roles in understanding the overall usability and UX.

Success rate of a task is defined as the percentage of users who are able to successfully complete the task. It is one of the most commonly used metrics in usability study.

Task duration is the time a participant spent to complete a task and measured in seconds. **Error rate** is a straightforward metric that indicates how often participants make mistakes while completing a task and it provides a better understanding of how much the testing target is usable.
In traditional usability testing **Satisfaction rate** is an important dimension and measured by questionnaires at the end. In this usability testing platform, it is measured by monitoring the facial expressions of the participant.
Gaze reply is the track of a participant's looking position coordinates on the screen. **Gaze heatmap** is the eye fixation of all participants.
Age and gender groups shows the clusters of participants for different gender and age range.
Scroll depth shows how much a participant scrolled on a specific page and measured in percentage.

	Data Collection Methods	Obtainable Measures	
Usability	Successfully completed tasks	Success rate	Entire workflow, Sentimental feedback, Concentration on the task, Visibility/understandability/adequacy of contents (information provided such as title/paragraph), Visibility/comprehensibility of commands
	Failed tasks	Success rate	
	Task completion time	Task duration	
	Errors occurred during task	Error rate	
	Satisfaction questionnaire	Satisfaction rate	
Insight	Gaze tracks	Gaze replay	
	Facial expression detection	Emotional state	
	Age detection	Age group	
	Gender detection	Gender group	
Engagement	Gaze tracks	Gaze heatmap	
	Mouse clicks	False clicks	
	Scroll	Scroll depth	

Table 3 Usability platform measures and data collection methods

Once the usability test conductor defines the tasks for the participants, the platform can start capturing all the defined measurements listed and visualizes the results on the platform dashboard.

4.1.1 Database structure

In order to store the related measurements and collected data, a relational database is designed. Each participant is given a unique id in order to be differentiated from other participants. All the measurements and task id he/she is performing at the moment is stored separately in the database.

The database structure is presented as tables in the following:

Emotions									
userID	joy	surprise	disgust	sadness	anger	fear	neutral	timestamp	url

Fixations				
idFixation	x_coordinates	y_coordinates	requestID	timestamp

Interaction								
userID	url	viewportWidth	viewportHeight	clickX	clickY	scroll	timestamp	scrollDepth

Participants		
userID	timestamp	requestID

Tasks				
userID	taskID	timespent	note	timestamp

Users				
userID	age	gender	timestamp	requestID

Table 4 Relational database structure

The explanation of some data fields from the data base:
userID: unique identifier of the user who is performing the task
url: the url of the page active at the moment of data capture
timestamp: the timestamp of the happened event
idFixation: id of gaze tarmacking data received
x_coordinate: x coordinate of where the user looked at when the data capture occurred
y_coordinate y coordinate of where the user looked at when the data capture occurred
viewportWidth: width of the viewport size user used to visit the page
viewportHeight: Height of the viewport size user used to visit the page
requestID: id of the requests sent from the client
scroll: whether the scroll event is triggered or not
clickX: x coordinate of where the user clicked/tapped on the screen
clickY: y coordinate of where the user clicked/tapped on the screen
scrollDepth: percentage of page scroll at the time of data capture
taskID: unique identifier of tasks defined
timespent: the time spent in completing a specific task
note: notes written by users during the tasks
age: age of the user
gender: gender of the user

4.2 The System Architecture

A centralized client-server architecture is designed and implemented concerning the hardware requirements to host the deep learning models and also to simplify the test procedures for the participants. The system based on deep learning algorithms, processes huge amount of data mainly in the form of image. When the different types of participants' devices are considered, it is near to impossible to guarantee the reliability of the system, it is also possible that a participant simply doesn't have a powerful device to run the system.
In client-server architecture, many clients can communicate with a centralized server via computer network, send and receive requests from the server. A client can ask the server to perform powerful computations and return only the results. All the hardware requirements to run a resource intensive program is also centralized on the server side.
Centralized architecture is also best suited for the remote usability platform, since all the test data has to be collected and processed together to conclude the results. A database can be hosted on the server where all the test data is stored together inside the database. It simplifies the data analysis process as well.

Considering the accessibility of the platform, almost any modern-day laptop or mobile device equipped with a frontal camera is enough for a user to participate the usability test from anywhere around the world with an Internet access.
It also has a big advantage in cost effectiveness compared to other types of architectures. The cost of maintenance is low.
The whole system for pc consists of following 4 different modules, implementation details of each modules are explained separately in the following paragraphs.
Client side of the system contains web plugins developed with JavaScript. Main functionalities of the plugins are capturing user interactions, i.e. interactions timestamps, clicks and scroll coordinates, the webcam handling. How and when to activate the webcam, capturing image frequency features are the most important ones, because both gaze tracking and emotion recognition modules need the facial image of the participant that is captured by the device camera. These images then encoded in base64 with encoder function and sent via HTTPS POST request to the server.
Server side of the system is developed Flask – a lightweight WSGI web application framework. It is one of the most popular Python web application frameworks. The server listens to the requests sent by clients, once a request arrives on the server, the request is decoded, parsed to extract all the original data then saved to the database. In order to prevent any data loss on the server side, a Redis caching layer is implemented. The trained face detection, facial expression detection, age and gender detection, gaze tracking models are hosted on the server.

- Face detection
  - Dlib is used to detect one or more human faces in a frame and provide main face landmarks coordinates.
- Facial expression detection
  - The trained VGG network model has been defined to take 64x64 pixels face images in grayscale by the input layer, and returns the Ekman's Emotions (joy, surprise, anger, disgust, sadness, fear and neutral) classification probability by the output layer
- Age and gender detection
  - The trained WideResNet network model is fed with 256x256 pixels aligned face images, it returns the classification class name Male or Female and age class name ranging from 0~101.
- Gaze tracking
  - The trained gaze tracking model takes in input the cropped images of left and right eyes separately and detected face image cropped from the original frame and a binary mask (face grid) used to indicate location

and size of the face within the frame, outputs the x-y coordinates of the eye fixation

The detailed training processes of these models are explained in the following sections.

Database is the data storage module used to save the outputs of the models and the user interaction, event timestamps, current task names, time spent on completing a task etc. MySql relational database is used to store all the data. It is the most secure and reliable database with high performance and on-demand scalability.

Dashboard module is an important module with the features to visualize the final result of the usability test. It also has a section dedicated for providing users with the task descriptions, a comment section if users want to express their concerns and problems occurred during completion of a task. it also tracks the task completion time. Multiple graphs and charts are generated on the different sections of the dashboard addressing different aspects of the usability. Tasks, Usability Analysis, Insights and Engagement are the 4 different sections inside Dashboard. All of them except the Tasks section are the 3 different aspects of the target platform analyzed

Detailed descriptions of each section and their functions are explained in the next chapter.

Data flow of the usability platform between different modules are depicted on the following architecture description. Data flow starts from the client devices, passed onto the server module, after processing the data it is saved inside the database and red by dashboard module when the results of the tests are required.

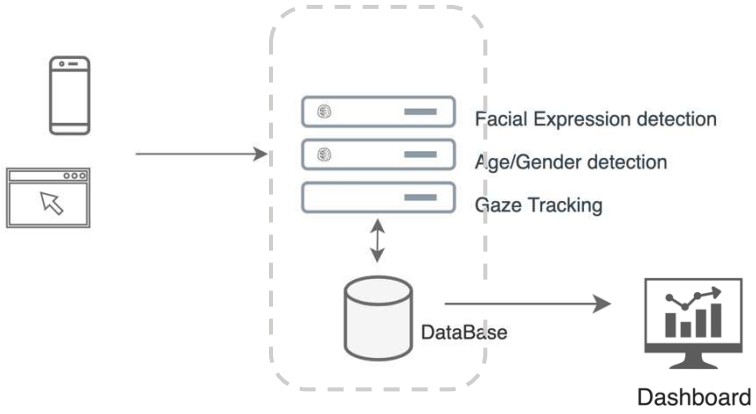


Figure 31 System architecture

4.2.1 Mobile SDK

To fully support UX assessment of mobile apps in the wild, MoBeTrack provides users' demographic data (e.g. age and gender), performance data (e.g., time to navigate a screen) and usage data (e.g., scrolling, tapping). Moreover, it exploits eye tracking and emotion recognition systems to allow the collection of behavioral information.

The system makes use of a centralized architecture that, as shown in Figure 1, has two main actors: the iOS SDK on the client side, and the Deep Learning (DL) platform on the server side. The mobile SDK is an iOS framework that exposes some APIs to monitor all the user interactions during the mobile app usage. Among these features, there is the possibility to activate the camera that takes different photos with a certain frequency.

These photos are Base64-encoded and sent to a server by HTTPS protocol. The central server that supports all the platform architecture, handles incoming calls from the iOS framework through a REST interface developed in Python, that waits for POST HTTPS calls addressed to the exposed endpoint. Once the call is received, the content is parsed and decoded to get all the data, included the original JPEG subsequently stored on the physical memory.

After that, the JPEG file name, that uniquely identifies the photo, is stored in three different Redis queues so that, through the path of the directory where the files are physically located, it is possible for every DL Tracker module to obtain the position of the photos every time it arrives. These queues are so used by the three Tracker modules to respectively obtain the estimation of the user's gaze x-y coordinates, his emotional state, and the gender and age information. All of DL Tracker modules are based on Convolutional Neural Networks (CNN) implemented in Python. Whenever the processing of a photo is terminated for all CNNs, the resulting data will be stored in a database and the photo itself will be permanently deleted from the server. All the stored data will be available through an Analytics Web Platform.

The client-side of MoBeTrack is a framework designed for Apple smartphones. This framework, once embedded in iOS applications, allows to send to the server all the activities performed by the user with the app. Firstly, by using the smartphone front camera, it takes photos silently with a frequency of 0.5 Hz and analyze them at the server-side. Secondly, each time a user taps the screen, it stores the x-y coordinates of the tapped point (in pixels).

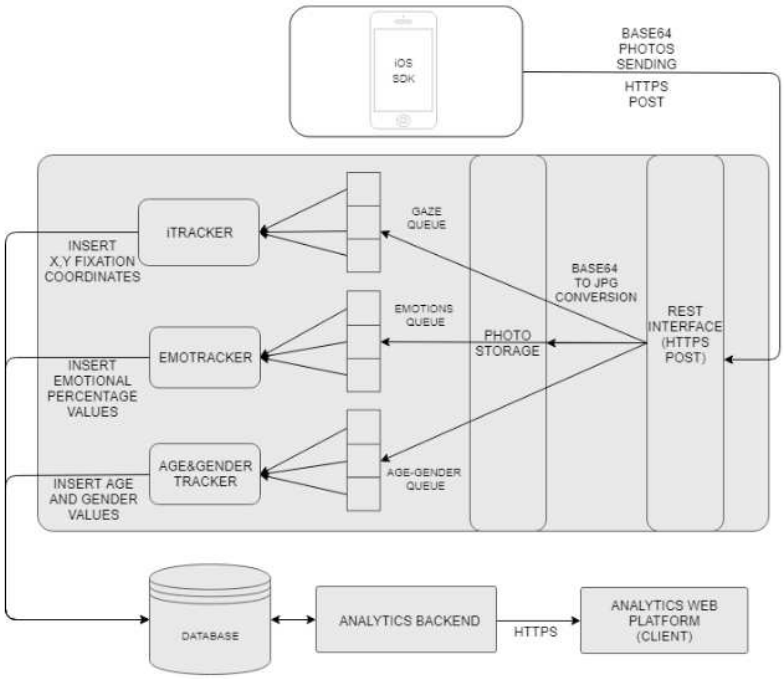


Figure 32 Platform architecture

Also, information about scroll activities is taken into account. Each time the user scrolls the current view, the y-offset from the left-top corner of the screen is updated. All these values are sent to the server whenever one of the aforementioned actions is performed.

The android version of the SDK has the same functionalities as iOS version. It is developed with Java programming language that has the cross-platform advantage; it can run on any android device.

One challenging functionality is the implementation of taking photos in the background as a service on the device without the camera preview. Users are asked the grant the camera access when they open the web application specifically designed and developed for the usability testing purpose. A web view is embedded inside the android application that automatically navigates the user to the predefined destination once the application is opened.

Android version of the SDK is also developed using Java. Main classes and their functionalities are explained in the following paragraph.

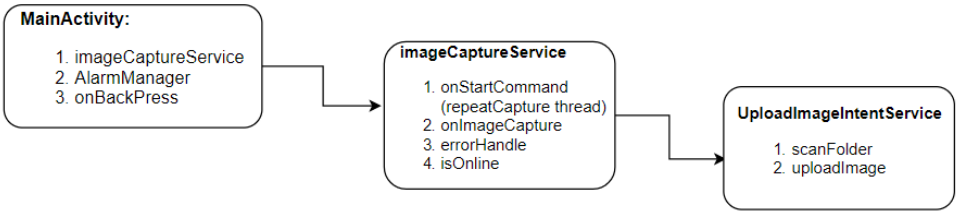


Figure 33 Main classes implemented on the Android version of the SDK

- **MainActivity**
  - imageCaptureService initiates a background service that takes pictures repeatedly using the front camera of the user.
  - AlarmManager set the time period of the repeating task, periodically call the service to take a picture.
  - onBackPress if the app has previous activities, it goes back to the previous activities. If no previous activities to go back, it frees the camera and quit.
- **ImageCaptureService**
  - onStartCommand starts the service, it creates a thread called repeatTakePicture and takes a picture.
  - onImageCapture returns the photos taken, creates local database and records the information. It also checks the internet connection and starts an intense service that uploads the photos to the server.
  - errorHandle is triggered if there is any kind of error happens.(ie. No front camera, front cam is not free etc.)
  - isOnline checks the Wifi internet connection of the phone and returns the value;
- **UploadImageIntentService**
  - ScanFolder scans the folder to check if there any photos that not uploaded.
  - uploadImage if there are photos that didn't uploaded yet, it checks the WiFi internet connection and uploads the photos.

An important implementation detail on the android device is the difference between service and intent service defined on the android developers guide. Their difference is important in order to make the application behave correctly. Services runs on the main thread and it continues running until it is manually stopped. Intent service is also a service, but it creates a separate working thread and destroys itself when the task is finished. Therefore, intent service is better suited for the task.

The case study conducted on both iOS and Android devices and their results are stated on the case study chapter.

4.3 Development of the Platform

The client side of the system is developed in JavaScript. In order to test the usability of a website, the scripts can be simply embedded in the HTML source code of the target website. some canvases elements are required to be added to the HTML source code as well. It is needed for the face detection module. The face detection module used is *face-api.js* that built on top of *tensorflow.js* core which implements CNN to detect faces. Participants are asked to grant access for the camera when they are start the task. The camera periodically captures participants frontal face, resize the captured image to 640×480 and converts the image to Base64 string. Besides face image, the user interaction data such as coordinates of the clicks, scrolls etc. All the data collected is sent to the server with asynchronous https web requests once the url of the page changes.

The server side of the system is developed using python Flask framework. It listens to the client requests. Once it receives a request containing a Base64 encoded image, the image is decoded and processed with the emotion recognition, age and gender recognition modules. After extracting the emotion, age and gender information the image is simply deleted, the extracted information together with other interaction data are saved to the database.

The dashboard module not only visualizes the data saved during the test, but it also processes them to better reflect all the platform features defined previously. The dashboard consists of three separate sections usability, insights and engagement. Each section has graphs related to the section. Besides them, there is a task section, where all the tasks are listed. Each participant to the usability testing should first go to the tasks section to understand the instructions for how to conduct the tasks. All the available sections together with what's included under the section are listed below.

- Tasks
- Usability analysis
  - Completion rate
  - Time spent
  - Satisfaction rate
- Insight
  - User clusters
    - Age, gender
    - Geolocation
    - Devices
  - Heatmaps

        - ❑ Click heatmap
        - ❑ Scroll depth
    - Feedbacks
        - ❑ General view
        - ❑ Funnel
        - ❑ Emotion graph
- Engagement
    - Click path
    - Gaze path
    - Valence
    - Emotions

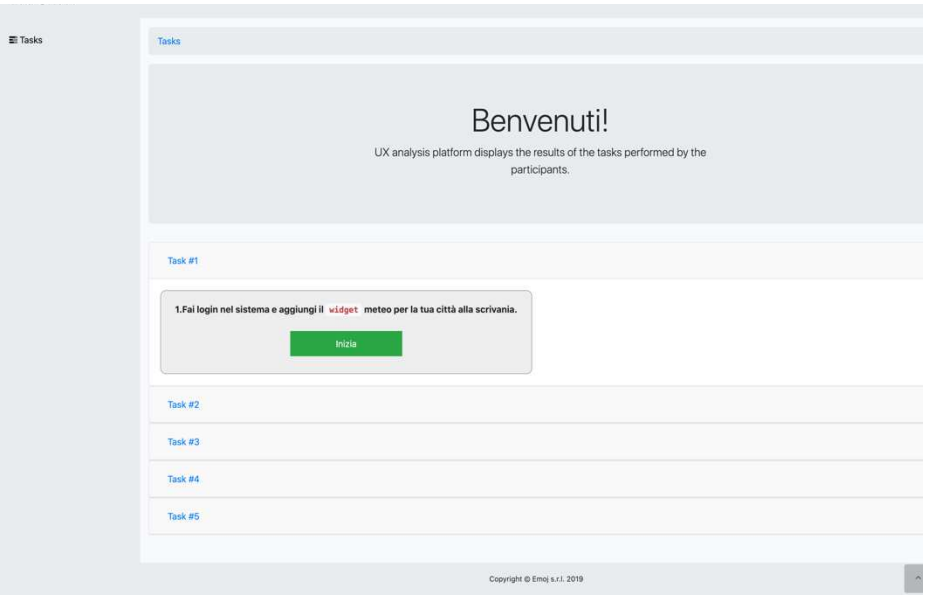


Figure 34 Tasks page

On Tasks page, the tasks are listed together with a start button. After carefully read the instruction, a participant can start the task by clicking on the start button. It brings the participant directly to the target site that needs to be tested.

On usability analysis page, there are Task completion, Task efficiency and Satisfaction rate graphs.

Task completion graph indicates how many tasks are present on the current usability test, and which tasks are completed by how many participants. For instance, form the

screenshot, it can be understood that there are 5 tasks defined in the current usability test and each task is completed by all 12 participants.
Task efficiency graph indicates on average, how much time is spent on each task in seconds.

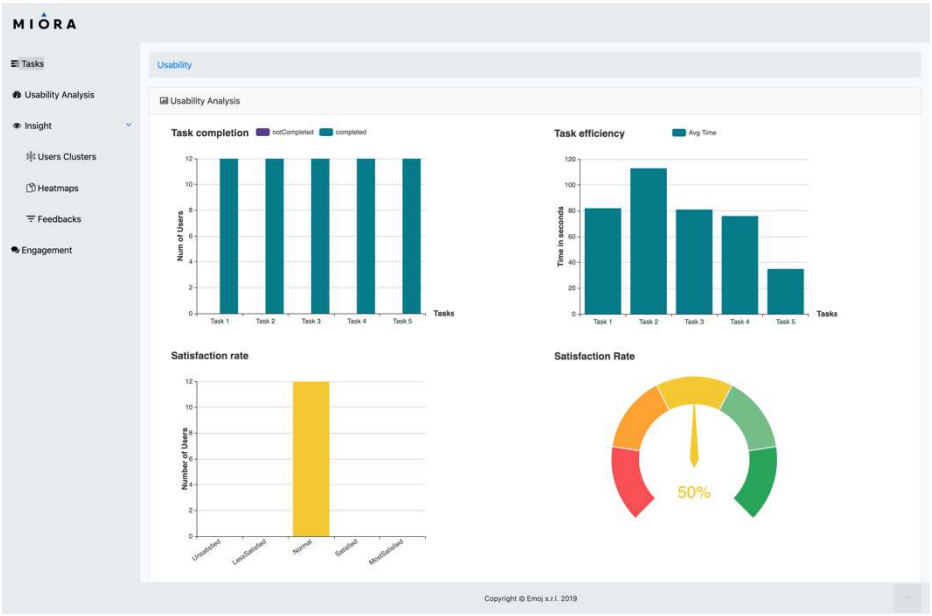


Figure 35 Usability Analysis page

Satisfaction rate is the average detected emotion from all the participants. It has 5 different categories Unsatisfied, Less Satisfied, Normal, Satisfied and Most Satisfied. Another satisfaction rate widget is also indicating the same results as the histogram.
On the Insight section, there are User clusters, Heatmaps and Feedback subsections. User clusters shows the overall statistics about the participant. They are clustered according to the devices they have used to complete the tasks, gender, age and geographical location. User distribution graph indicates how many percentages of the participants used mobile phone, laptop or desktop to accomplish the tasks. On age and gender distribution graph, participants can be filtered by gender to see the age distribution of the participants separately. Participants geographical distribution is visualized on the "Users in the World" graph. It is plotted based on the IP address of the device's participants used to participate the usability test.

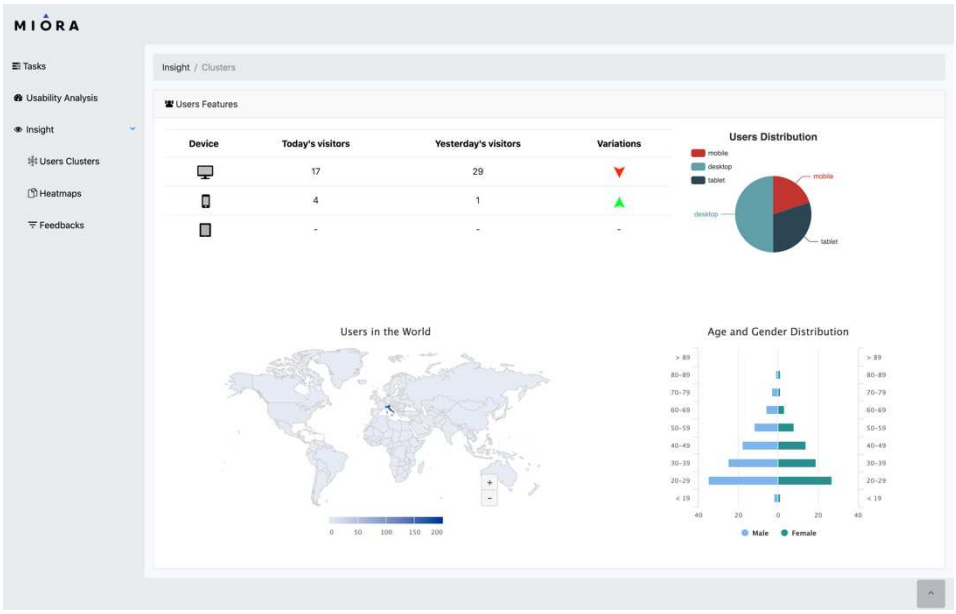


Figure 36 Users Clusters page

The Heatmap subsection has the interactions and gaze heatmaps. It visualizes the overall result of all participants, how often they have interacted or attracted to a specific part of the page. It provides important information on the errors that participants made during performing the tasks.

Feedback subsection is the most important part of the platform which provides many crucial analyses on the target platform. Page structure graph, Funnel and Interactions heatmaps related to the navigation paths, Average emotions' graph of all participants are present in Feedbacks subsection.

Figure 37 Heatmap subsection

Page structure graph is constructed according to the structure of the target platform. When a node is selected, a Funnel chart is drawn from the starting page to the selected page, all the interactions occurred between these two pages, convention rate, valence, interactions heatmap and emotions captured are plotted. If it includes multiple pages, the interactions heatmap for each page is constructed in a gallery slider format.

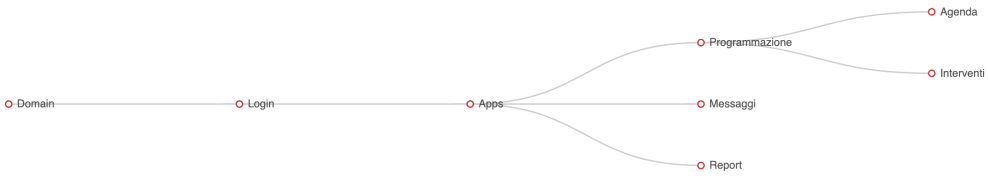


Figure 38 Page structure graph

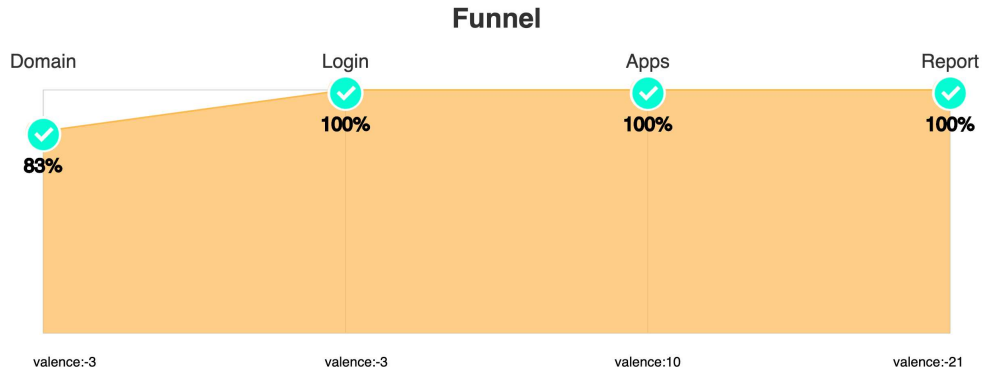
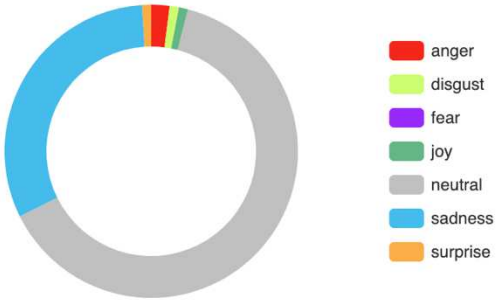


Figure 39 Funnel graph

Each vertical line on the Funnel graph represents a page present in the target platform, the percentages indicate how many percentages of the overall participants actually visited the corresponding page. For instances, if the total number of participants are 13, the Domain page is only visited by 10 participants. The valence indicates total positivity or negativity expressed by the participants while visiting the page, the range of valence is from -100 to 100.



Average emotions graph visualizes the average emotions of all participants who visited

Figure 40 Average Emotions graph

the selected pages.

On the last Engagement section, Click and Gaze path graphs, valence graph and emotion graph correspond to individual participant are present.

Valence graph tracks the positivity or the negativity that a participant expressed while accomplishing a task. It can be seen as the indication of satisfaction a participant expressed.

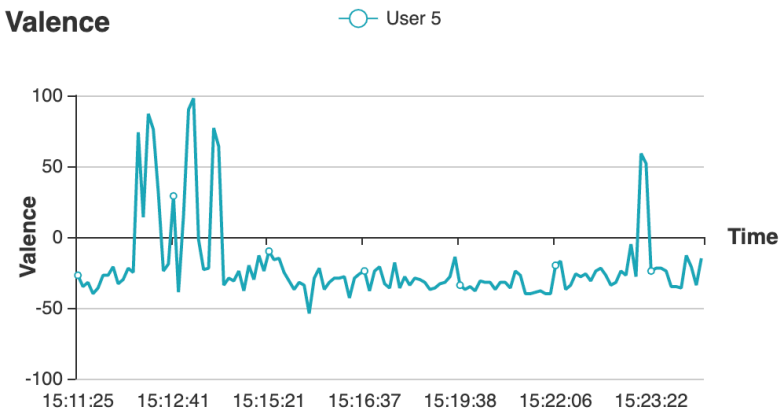


Figure 41 Valence graph

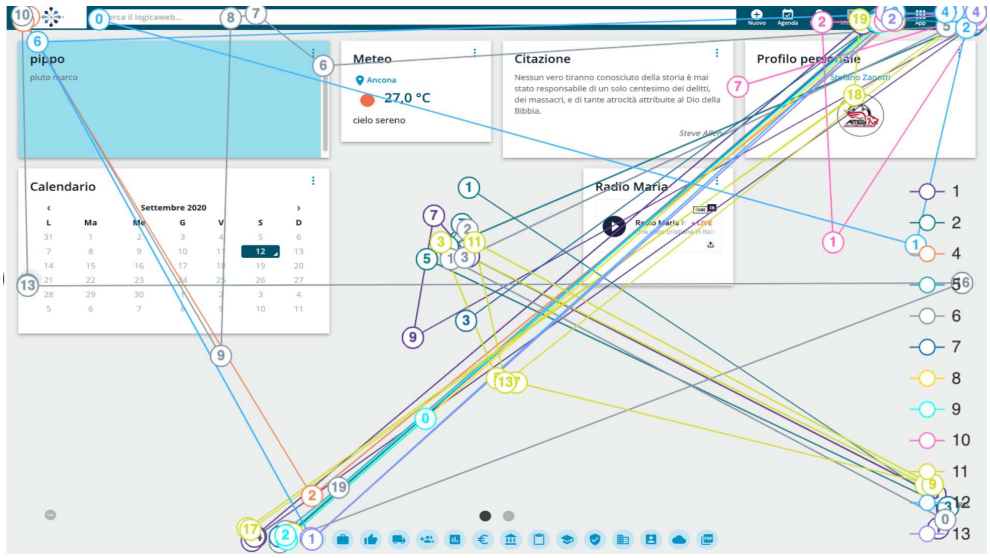


Figure 42 Click path

The Click path graph visualizes all the clicks performed by the participants in the occurring sequence and differentiates each participant with different color. From the example, it can be understood that there are 13 participants in this usability test and all the click interactions are indicated for each individual.

Chapter 5. Case Study: Usability Analysis

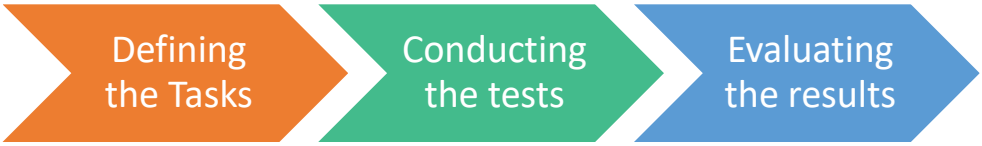
In order to validate the functionalities and usability of the usability analysis platform in the real-world scenario, a case study is conducted to obtain preliminary results on the SNAPSI website platform with the collaboration of FLOWING s.r.l and SNAPSI s.r.l. SNAPSI platform is an enterprise multi-functional management platform with many complex features.

Assessments of the interfaces, functionalities, adequacy and effectiveness of the platform are evaluated in the case study. Important preliminary qualitative and quantitative assessments are obtained at the end of the case study. It forms the base of the more detailed large scale quantitative assessment in the future.

Relatively simple tasks are defined for the small-scale usability test participants with different backgrounds. Some expert users are chosen from the developers of the SNAPSI platform. The data they generated while performing the tasks are considered as references. Some other participants who are the potential users of the platform also invited to the usability testing to evaluate the usability of the SNAPSI platform.

During the test, each individual's interaction with the platform is recorded. These recordings then are compared to the data generated by the platform to validate the effectives of the methods implemented in the platform. All the participants are asked to conduct the usability tests in full screen in order to ease the click coordinates validation. All the deep learning models embedded in the platform are previously validated at the end of their trainings, therefore validation phase of the case study is more focused on the user interactions such as click positions, navigation paths etc.

Usability testing workflow is defined in the first step.



5.1 Definition of the Tasks

Five tasks are defined prior to the usability testing evaluation. They are listed below:

1. Logging into the system and add a weather *widget* of your current city to the desktop of the platform.

2. Login to system, go to Programmazione, navigate to Agenda by selecting it and assign a job to the colleague *Ideato*.

3. Login to system, go to Programmazione, navigate to Interventi, change the layout of the grid by selecting *Minore di oggi*.

4. Login to the system, go to *Messaggi*, send a message/attachment to *Flowing*.

5. Login to the system, go to *Report*, delete the first report on the list.

Figure 43 Tasks defined

5.2 Conducting the Tests

After finishing the definition of the tasks, people with different background knowledge have been invited to participate the usability test.
Total number of participants of the usability testing of SNAPSI platform is 12. One of them is the expert user of the platform. The tests are conducted in remote moderated usability testing manner. All the processes of the participants performing tasks are recorded in order to manually evaluate the tasks. Some requirements and conditions of the usability testing platform is explained to the participants before they start the test. the requirements are:

1. They should perform the task in a relatively illuminated environment with a pc that has a camera.
2. They should start each task by pressing the start button below the task description and close the opened tab once they finish a task and click on the

button for informing the system, they have finished the task. This button starts and stops a background timer that keeps track of the task duration.

3. The participants are asked to leave some notes or comments according to their experience after finishing a task and press the send button to send the notes.

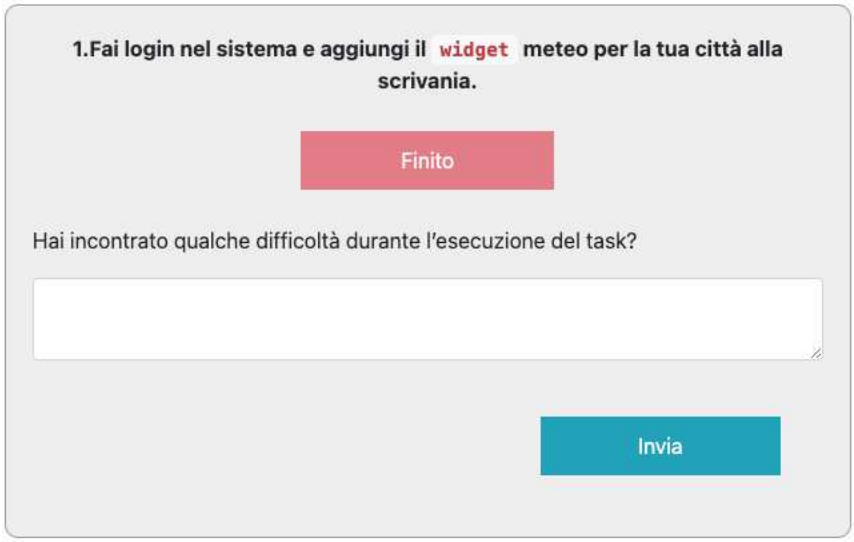


Figure 44 The note/comment section

The comment and the send buttons appear only after clicking on the finish button.

5.3 Evaluation

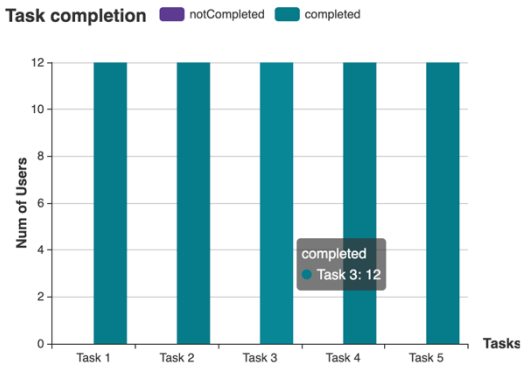
Before analysing the collected data, it is important to check their correctness. It is achieved by watching the test recordings and comparing the user behaviour to the plots generated with collected data.

Because tests are conducted in full screen mode, using the screen resolution of each individual participant a ratio between their screen resolution and full HD (1920*1080) resolution can be calculated. Using this ratio, all the click coordinates can be scaled up or down then be plotted on top of each web page screenshot image.

This method greatly reduces the complexity of comparing each individual click coordinate to the video recording.

The results suggest all the user interactions captured during the test are correct. In the next phase, analysis on predefined tasks is conducted.

Completion rate of each task:



1 Task 1: 100%

2 Task 2: 92% (User9)

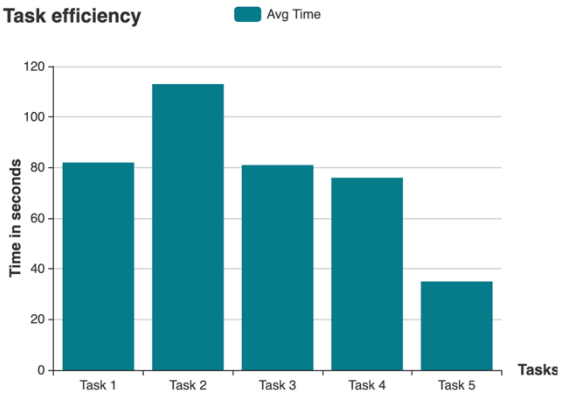
3 Task 3: 92% (User12)

4 Task 4: 92%(User13)

5 Task 5: 100%

By analyzing the results generated by 12 participants, task 2, 3, 4's completion rate is less than 100% which indicates there are some participants couldn't be able to finish the tasks.

Time spent by expert user:



1 Task 1: 41s

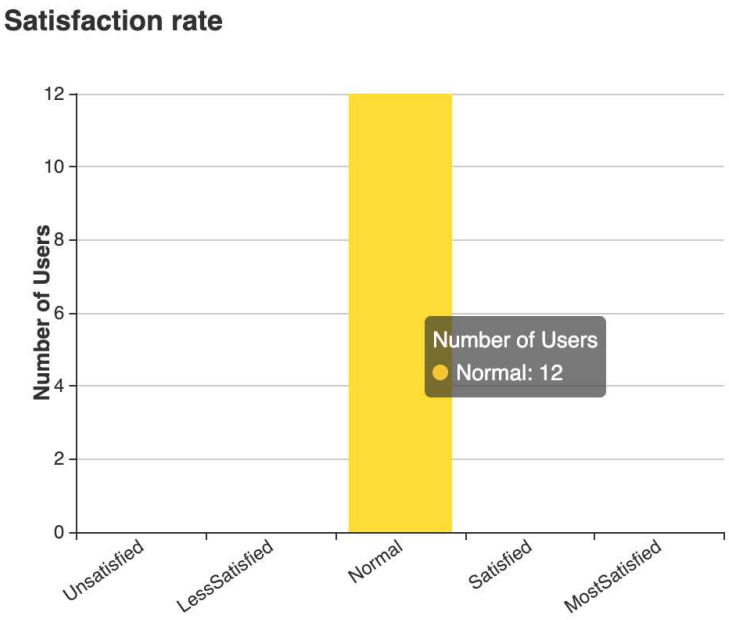
2 Task 2: 44s

3 Task 3: 24s

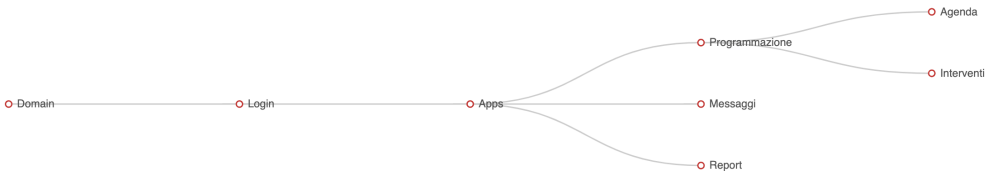
4 Task 4: 63s

5 Task 5: 32s

By comparing the time spent by expert on each task to the Task efficiency graph it can be seen that the average time spent on task 2 and task 3 have huge difference.



The satisfaction graph shows that the average feedback from 12 participants is normal. It can be further analyzed by looking at the valence graph for each individual participant. The hierarchical tree graph shows the structure of the pages that involved in the test, it also roughly suggests the paths that the participants took in order to complete the test. The first funnel chart corresponds to the task 1, adding a widget to the main page, the second one corresponds to the task 5 – deleting a report from the report list. This result suggests we have 100% conversion rate on these two test.



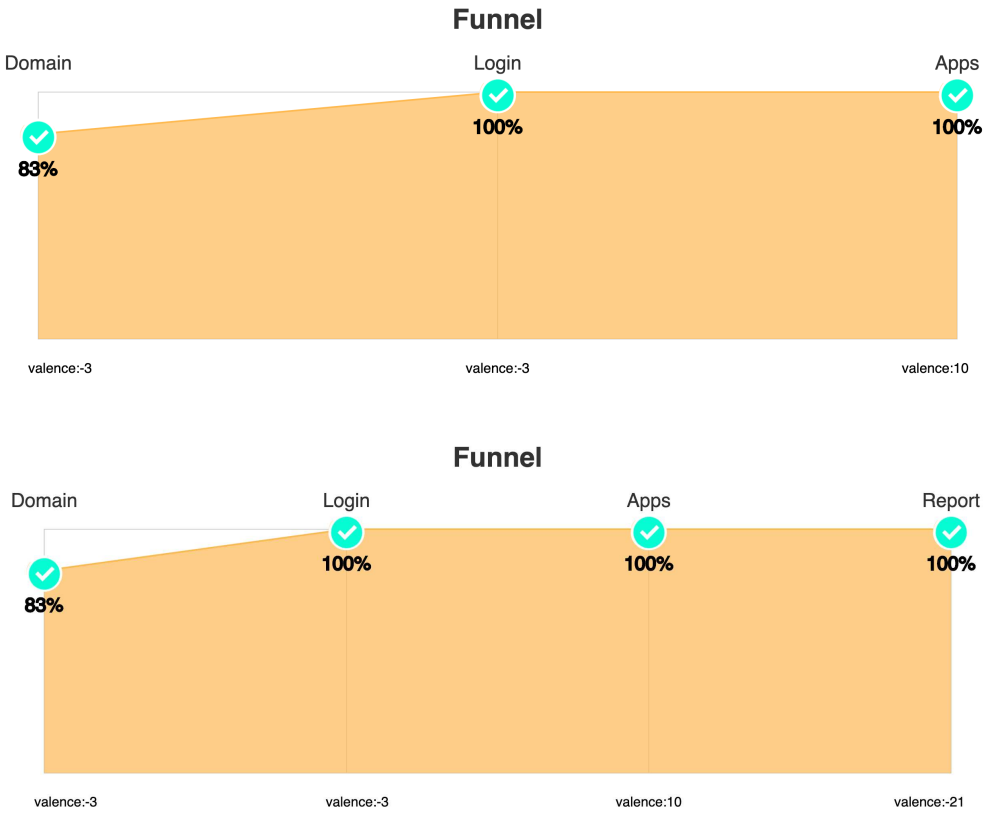
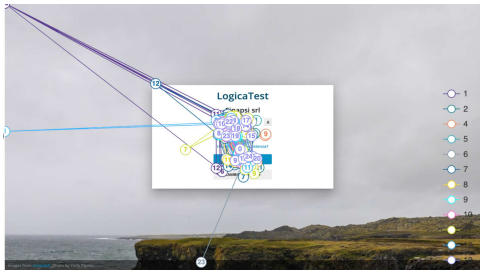


Figure 45 Conversion rate of Task 1 and Task 5

Here are all the interactions occurred on each page for each participant.



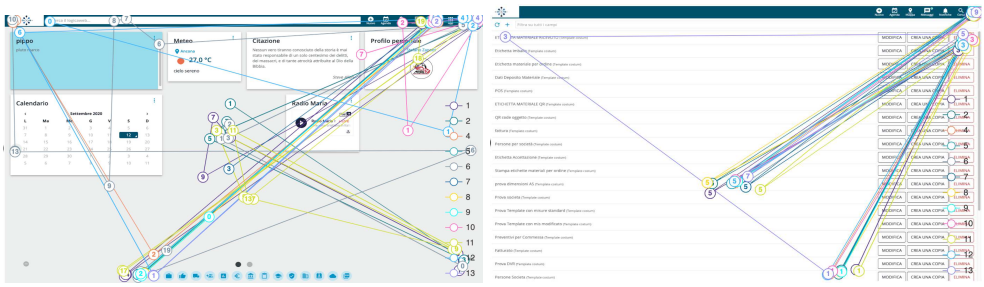


Figure 46 Click paths of Task 1 and Task 5

These funnel charts correspond to task 2 assigning a new task to *Ideato* in agenda page and task 3 – changing the layout of the grid in *Interventi* page.
We can see some drop down on *Programmazione* and Agenda pages. This is because Agenda page has different URL's depends on which navigation path a participant has taken.

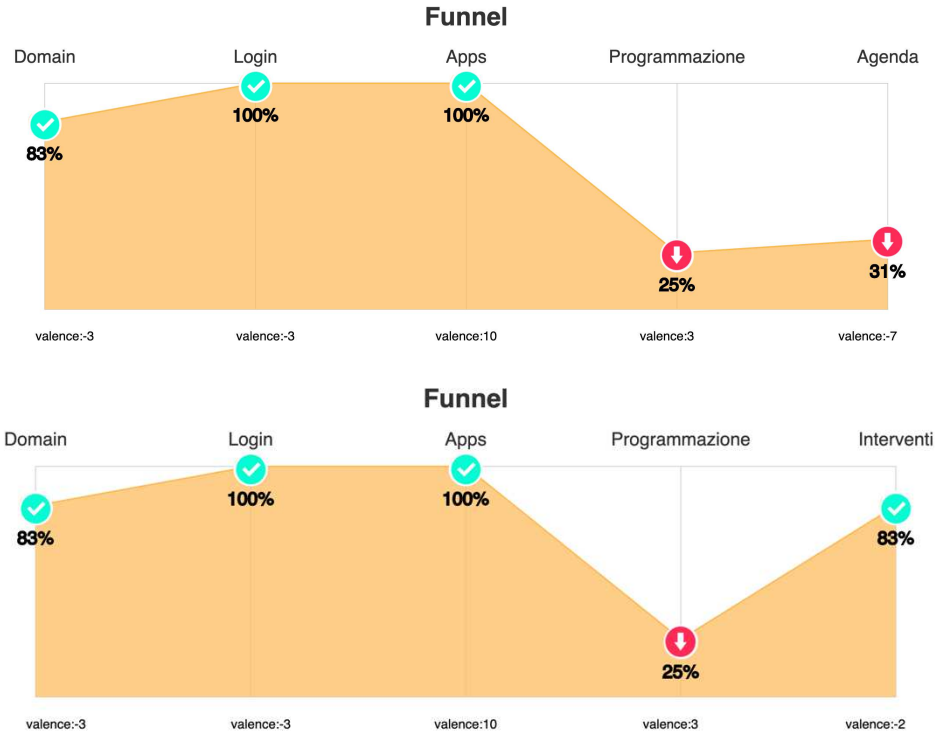


Figure 47 Conversion rate of Task 2 and Task 3

Expected path:

Log in →Apps → Programmazione → Interventi → Agenda

Alternatives:

Log in → Apps → Agenda

Log in → Apps → Programmazione → Agenda

Same thing happened to for *Interventi* page as well, some participants finished the task on *Programmazione* page and one of them was failed to finish.

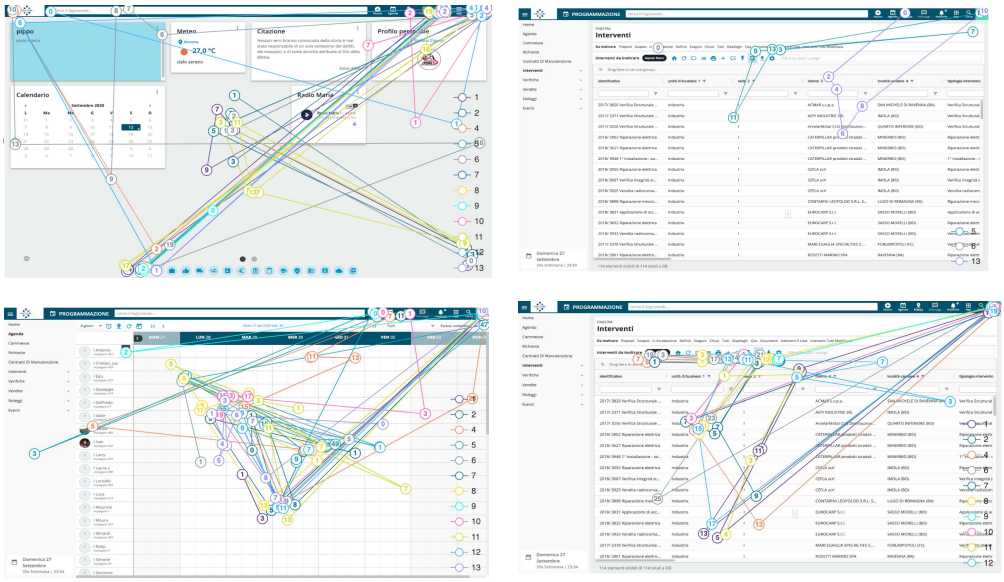


Figure 48 Click path of Task 2 and Task 3

One participant was not able to finish the task 4.

Figure 49 Funnel graph of Task 4

92% conversion rate on *Messaggi* page demonstrates the previous claim.

Figure 51 Click path of Task 4

The overall emotional feedback from the participants indicates they have experienced

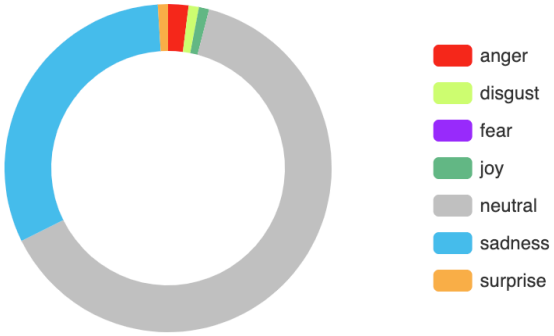


Figure 50 Overall Emotional Feedback

some sadness during the test. this is due to the fact that some participants were unable to finish the task 2 and task3. It was very difficult for them. The valence graph also demonstrates the negative emotional status they have experienced.

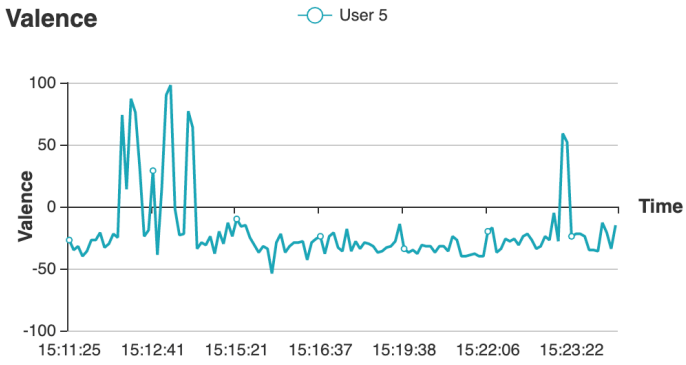


Figure 52 Valence graph of User 5

User 5 experienced difficulties while completing the task 2 and task 3. The valence graph correctly reflects the user 5's experience from the test. User 10 easily accomplished all the tests except for task 3. It can be seen from the Valence graph that User 10's facial expressions changed to express more negativity while he or she is completing task 3.

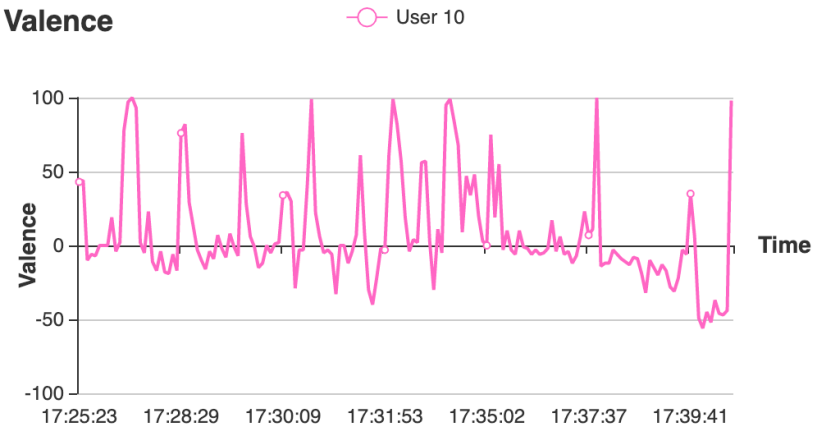


Figure 53 Valence graph of User 10

In order to provide a comparison to Valence graph of User 5 and User 10, User 2's Valence graph is also presented.

User 2 accomplished all the tasks quickly and didn't experienced any difficulties during the test.

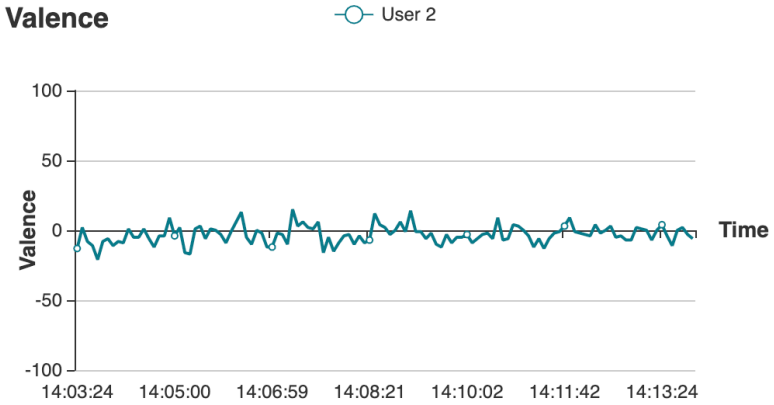


Figure 54 Valence graph of User 2

5.4 Results

Most of the users experienced some difficulties on task 2 – creating a new task for *Ideato* and task 3 – changing the layout of the grid.

Figure 55 Agenda page with the indication of usability issue

The *Nuovo* button on the page caused some confusions to some participants when they are completing Task 2.

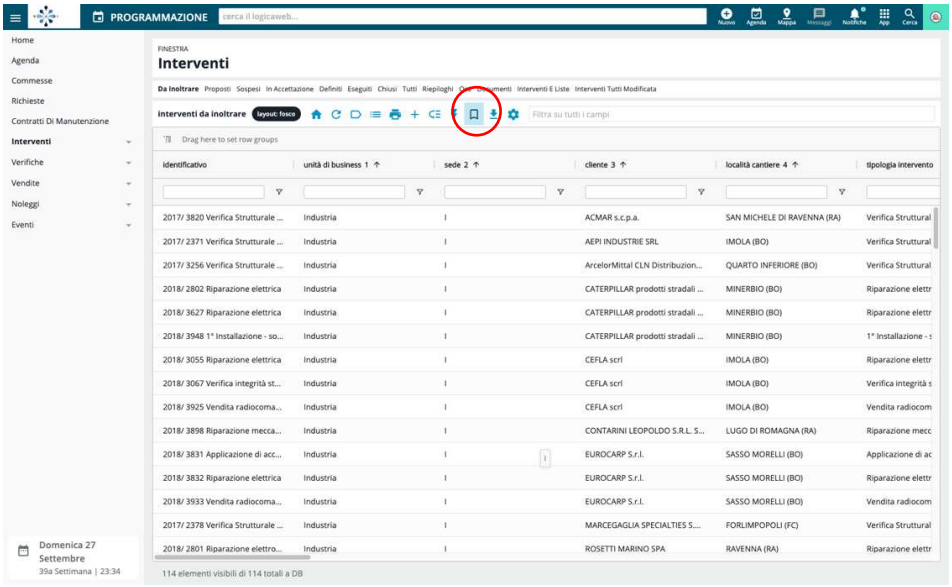


Figure 56 Interventi page

Almost all the participants were unable to recognize the bookmark icon and relate it with "saved layouts". Even though this caused some usability problem for the participants it cannot be treated as a big issue since it is easy for them to remember once they are told. Some other problem occurred during completion of task 3 are:

- The recipient is selected only by a checkbox before sending the message, which seems confusing to some participants.
- There is no "send" button on the page the only way to send is by pressing the "Enter" key on keyboard.
- Some participants reported that there is no direct log out button on Message site.

Figure 57 Messagi page

5.5 Usability Testing on Mobile devices

MoBeTrack – the toolkit to analyze user experience assessment of mobile applications previously introduced. The system is able to collect, analyze and visualize data about user interactions/behavior with mobile applications supporting UX designers and usability test conductors.

It is used to conduct preliminary tests on iPhone 4, iPhone 6s and iPhone 7 Plus to determine CPU and battery consumption and required RAM allocation, by varying the

screen brightness and the frame shots frequency. Results are reported in the following table (Generosi et al., 2019).

Shots frequency	Screen brightness	iPhone 4			iPhone 6s			iPhone 7 Plus		
		CPU (%)	RAM (MB)	Battery (%)	CPU (%)	RAM (MB)	Battery (%)	CPU (%)	RAM (MB)	Battery (%)
0.5 Hz	50%	25%	10	11%	7%	21	7%	14%	30	4%
	100%			16%			15%			13%
1 Hz	50%	28%	6	12%	9%	15	9%	19%	38	5%
	100%			17%			15%			14%
3 Hz	50%	56%	9	12%	15%	21	10%	24%	90	7%
	100%			19%			18%			15%

Table 5 Test Results

It can be observed that the task is resource intensive. It is better to use more powerful devices during the test.

The android version of the system is tested on Xiaomi Redmi 3 Pro, Xiaomi Mi 4c, Samsung Galaxy S6, Meizu M3 Note and LG G3 smartphones. Similar results are emerged from the test.

A new type of on device embedding of the trained CNN models are experimented on android device. Age and gender detection model is converted to TensorFlow Lite model and directly embedded inside the mobile device to evaluate the performance. The experiment is conducted because some users may not want their facial images are sent to a server even if it is not stored there. However, as it is expected at the time, the processing power of a mobile device is limited, and the speed of extracting information needed form the images taken is not ideal.

If all the models are successfully embedded on the device, the calculations are done on local device. Only the results extracted from the images are sent to the server to be stored on the database.

Assuming the device is powerful enough to run the system, another disadvantage of this type of implementation could be encountered - the energy consumption of the device. Since all the process are completed locally on the device it needs more power to complete the required operations.

5.6 Discussions

Case study results demonstrate the effectiveness of the new remote usability platform. The hidden designing flaws are successfully identified by the system. How the users are interacted on the target site is correctly captured. The emotion recognition results also

match the experience of the test participants as it is reflected on the valence graph. Effectiveness, efficiency and satisfaction matrices of usability and user insights, engagements are correctly addressed by the platform based on the small-scale usability test. Heatmaps, histograms and funnel charts made it very easy for the usability experts to understand the test results and quickly identifying the hidden designing and development errors. This is particularly important because it doesn't require further manual analysis like other remote usability testing platforms to make the final conclusion.

The mobile SDKs also show the feasibility of the approach, it could make a big difference in remote usability testing field. It is preferred to deploy the resource intensive computations such as face detection and emotion recognition, age and gender detection are conducted on the server side, in order to guarantee a good user experience on the mobile devices.

When the accuracy of the test results is evaluated, it can be seen from the plots that the user interaction captured by the system is accurate. However, some bugs have been encountered, identified. Mismatching's of the page screenshot and interaction plotted happened on the final task. All the spot 1 circles should be mapped on the previous page not on the current page. However, it is a programming bug that can be easily fixed.

Figure 58 Task 5 result

The gaze tracking model prediction results on pc is not as good as expected, it is mainly because there is no huge amount of labelled data available to train the network. The performance evaluation study on the gaze tracking model also suggests its accuracy can be improved further. The data collection process using the developed web application is

active, once enough data is collected the gaze tracking model will be retrained to produce better results.

Based on the results of the preliminary small-scale usability test conducted on the platform, it strongly suggests that such platform has enormous potential in usability testing industry. The most important advantage of this platform is its cost effectiveness. It can hugely reduce the cost of conducting remote usability tests and UX assessments. There is no special equipment needed, the human resource and time to analyse the data collected is also saved.

Chapter 6. Conclusion

UX and usability testing has come a long way today, from the early days of spending huge amounts of money on recruitment, setting up testing labs and investigating huge amounts of time completing these studies.

In the present day there are various technologies available to help enterprises perform their usability testing on a large scale, across devices and user demographics at a faster pace, and in an economical manner. The implementation of artificial intelligence in the usability testing and improving UX is an innovative way of enhancing the usability of a product.

Some challenges encountered during the design, development and implementation phases are:

1. Face detection
    1) Frontal face detection is the first important step and crucial for obtaining reliable results from other modules of the system. Many datasets available includes face images taken from various different angle. After training a face detection model using these datasets, the model detects faces that are not frontal which causes inaccurate emotion recognition, age and gender recognition results. Dlib's HoG frontal face detection model is used to overcome this challenge.
    2) At the same time Dlib's frontal face detection algorithm is more sensitive to illumination which constrained the system to be used only in relatively well illuminated environment.
    3) Different face detection algorithms and models are available for configuring the system according to the environment as well. This implementation however is more challenging in terms of the development of the system.
2. Emotion recognition
    1) First challenge in the emotion recognition (Facial expression recognition) is finding a dataset with relatively high label accuracy. There are not many available datasets for some specific emotion category in Ekman's universal facial expression model. The sample size for fear and disgust categories are small and for that reason, the detection accuracy for these categories is not very high and reliable.
    2) Finding the best neural network architecture for facial expression recognition is a challenging and time-consuming study.

3) After integrating the face detection and emotion recognition, age and gender recognition modules, the system speed slowed down significantly, many experiments are conducted to find the bottleneck of the system and improve it to run in near real-time was also one of the challenges to overcome.

3. Gaze tracking
    1) Gaze tracking was the most challenging part of the entire system. There is no available dataset on gaze tracking. So, the process is started from developing an application to collect and build a gaze tracking dataset. The current results obtained is still not enough for a reliable gaze position prediction. However, it shows very promising results, and it is believed that by training the gaze tracking network with enough data will give a high accuracy calibration free gaze tracking model in the future.
    2) The test results obtained on mobile devices are better than the test results obtained from PCs. It is because there is more data available for training gaze tracking model for mobile devices. Yet still the accuracy needs to be improved further.
4. Capturing the user interaction
    1) The user interaction such as click or tap coordinates are easy to detect and capture, the challenging part is the scroll event. Based on the target website structure, HTML div containers can be embedded one inside another. There is built in JavaScript event detector classes that can detect the scroll event, but the event listeners have to be defined first for these containers to capture the scroll depth coordinate. For finding the right scrollable container, it is necessary to check the CSS properties of all containers and define a scroll event listener for each of them. Or by comparing the viewport size with container size it is possible to ignore some relatively small containers which doesn't affect the target webpage structure.
5. Taking pictures without a preview
    1) This is also one of the most challenging tasks in the system. For android devices it is nearly impossible to take photos without a preview, in another word activating camera app at the background process and taking pictures periodically while a user is performing the useability task. After many attempts and researching on the issue, IntentService class is used to overcome the issue.

6. Development
    1) Developing such a platform is also very challenging, different platforms, multiple programming languages for each platform for instance, Java for Android, Swift for iOS, Javascript and HTML, CSS for desktop and dashboard, Python on deep learning model training and server development.

After so many challenges, the platform is developed and evaluated by the case study to show its effectiveness.

It is developed based on deep learning, state of the art face detection, age/gender recognition and facial expression recognition technologies are integrated inside the platform to obtain more information and use it in remote usability and UX assessment.

Results of the case study suggests the platform is able to capture and reveal hidden usability issues. The user satisfaction can also be investigated with the platform. From the case study conducted, by analyzing the graphs and comparing them with the actual user behavior, it can be concluded that they are matched, which suggests the system is effective and usable to conduct remote usability tests.

The system can also effectively capture the users' interactions, facial expressions and interpret them accordingly. It is easy to use, participants can participate the usability tests with a device such as pc, mobile or tablet that has a frontal camera. There is no need for other special equipment.

The test results are accessible instantly, there is no need to manually analyze captured data. UX designers can quickly test their prototypes and adjust design before the release.

Cost effectiveness, efficiency, scalability of the platform makes it more attractive compared to the other similar products on the market.

Besides above-mentioned advantages, modular design approach applied during the development makes it very easy to update or replace each module separately if needed.

The deep learning models are developing very rapidly, every now and then researchers publishing new approaches that superior to the previous one. It is important to train new models with more advanced approaches to achieve better results and keep the platform up to date. It is also important to retrain the face detection, age/gender detection and emotion detection models when more datasets are available to enhance the capability of these features.

There are some issues need to be improved in the future.

Training of the models requires further improvement. Because of the limited datasets available for us, some emotion categories such as disgust has very low detection accuracy. It can be improved by fine tuning the emotion recognition model on more accurate labelled dataset.

Gaze tracking models trained has high error rate, the reliability of the models is also low. It is important to recruit more volunteers to participate in the collection of gaze data. Once a good reliable dataset is constructed, it is believed that better results can be achieved.

The dashboard of the platform needs to be improved so that the remote usability testing whether it is moderate or unmoderated can be conducted. One thing noticed during the case study is that every participant has to be communicated about how to follow the instructions and the requirements such as lightning conditions of the environment they conduct the tasks, always use the main monitor of the device (some participants use second monitor) etc.

Finding a better way to communicate more effectively and efficiently will greatly improve the test conduction efficiency.

References

Agustsson, Eirikur, Radu Timofte, Sergio Escalera, Xavier Baro, Isabelle Guyon, and Rasmus Rothe. 2017. "Apparent and Real Age Estimation in Still Images with Deep Residual Regressors on Appa-Real Database." *Proceedings - 12th IEEE International Conference on Automatic Face and Gesture Recognition, FG 2017 - 1st International Workshop on Adaptive Shot Learning for Gesture Understanding and Production, ASL4GUP 2017, Biometrics in the Wild, Bwild 2017, Heteroge*, 87–94. https://doi.org/10.1109/FG.2017.20.

Ahn, Hyung Il, and Rosalind W. Picard. 2014. "Measuring Affective-Cognitive Experience and Predicting Market Success." *IEEE Transactions on Affective Computing* 5 (2): 173–86. https://doi.org/10.1109/TAFFC.2014.2330614.

Ahonen, Timo, Abdenour Hadid, and Matti Pietikäinen. 2004. "Face Recognition with Local Binary Patterns." *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 3021: 469–81. https://doi.org/10.1007/978-3-540-24670-1_36.

Barsoum, Emad, Cha Zhang, Cristian Canton Ferrer, and Zhengyou Zhang. 2016. "Training Deep Networks for Facial Expression Recognition with Crowd-Sourced Label Distribution." http://arxiv.org/abs/1608.01041.

Benitez-Quiroz, C. Fabian, Ramprakash Srinivasan, Qianli Feng, Yan Wang, and Aleix M. Martinez. 2017. "EmotioNet Challenge: Recognition of Facial Expressions of Emotion in the Wild." http://arxiv.org/abs/1703.01210.

Bruun, Anders, Peter Gull, Lene Hofmeister, and Jan Stage. 2009. "Let Your Users Do the Testing: A Comparison of Three Remote Asynchronous Usability Testing Methods." *Conference on Human Factors in Computing Systems - Proceedings*, 1619–28. https://doi.org/10.1145/1518701.1518948.

Chollet, François. 2017. "Xception: Deep Learning with Depthwise Separable Convolutions." *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017* 2017-Janua: 1800–1807. https://doi.org/10.1109/CVPR.2017.195.

Dalal, N., and B. Triggs. n.d. "Histograms of Oriented Gradients for Human Detection." In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 1:886–93. IEEE. https://doi.org/10.1109/CVPR.2005.177.

Dehghan, Afshin, Enrique G. Ortiz, Guang Shu, and Syed Zain Masood. 2017. "DAGER: Deep Age, Gender and Emotion Recognition Using Convolutional Neural Networks." *ArXiv*.

Detector, Face. 2018. "PyramidBox : A Context-Assisted Single Shot." *Eccv*. http://openaccess.thecvf.com/content_ECCV_2018/papers/Xu_Tang_PyramidBox_A_Context-assisted_ECCV_2018_paper.pdf.

F. M. EL-firjani, Naser, Ebitisam K. Elberkawi, and Abdelsalam M. Maatuk. 2017. "A Method for Website Usability Evaluation : A Comparative Analysis." *International Journal of Web & Semantic Technology* 8 (3): 01–11.

https://doi.org/10.5121/ijwest.2017.8301.
Følstad, Asbjørn, Effie Lai Chong Law, and Kasper Hornbæk. 2012. “Analysis in Practical Usability Evaluation: A Survey Study.” *Conference on Human Factors in Computing Systems - Proceedings*, 2127–36. https://doi.org/10.1145/2207676.2208365.
Frøkjær, Erik, and Kasper Hornbæk. 2005. “Cooperative Usability Testing: Complementing Usability Tests with User-Supported Interpretation Sessions.” *Conference on Human Factors in Computing Systems - Proceedings*, 1383–86. https://doi.org/10.1145/1056808.1056922.
Ganglbauer, Eva, Johann Schrammel, Stephanie Deutsch, and Manfred Tscheligi. 2011. “Applying Psychophysiological Methods for Measuring User Experience: Possibilities, Challenges and Feasibility.” *Human-Computer Interaction. INTERACT 2011 (Lecture Notes in Computer Science)* 6949: 714–15. http://www.springerlink.com/content/f7380g7k2179484t/.
Generosi, A., A. Altieri, S. Ceccacci, G. Foresi, A. Talipu, G. Turri, M. Mengoni, and L. Giraldi. 2019. “MoBeTrack: A Toolkit to Analyze User Experience of Mobile Apps in the Wild.” In *2019 IEEE International Conference on Consumer Electronics (ICCE)*, 1–2. IEEE. https://doi.org/10.1109/ICCE.2019.8662020.
Georges, Vanessa, François Courtemanche, Sylvain Senecal, Thierry Baccino, Marc Fredette, and Pierre-Majorique Leger. 2016. “UX Heatmaps.” In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, 4850–60. New York, NY, USA: ACM. https://doi.org/10.1145/2858036.2858271.
Gonzalez-sanchez, Javier, Mustafa Baydogan, Maria Elena Chavez-echeagaray, K Robert, and Winslow Burleson. 2017. *Affect Measurement : And Data Analysis*. *Emotions and Affect in Human Factors and Human-Computer Interaction*. Elsevier Inc. https://doi.org/10.1016/B978-0-12-801851-4/00011-2.
Gray, Wayne, and Marylin Salzaman. 1998. “Damaged Merchandise ? A Rewiew of Experinebts That Compare Usuabilty Evaluation Methods.” *Human-Computer Interaction* 13 (776111237): 203–61. https://doi.org/10.1207/s15327051hci1303.
Hassenzahl, Marc, and Noam Tractinsky. 2006. “User Experience - A Research Agenda.” *Behaviour and Information Technology* 25 (2): 91–97. https://doi.org/10.1080/01449290500330331.
He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. “Deep Residual Learning for Image Recognition.” *Enzyme and Microbial Technology*, December. https://doi.org/10.1016/0141-0229(95)00188-3.
Hollnagel, Erik. 2003. “Is Affective Computing an Oxymoron?” *International Journal of Human Computer Studies* 59 (1–2): 65–70. https://doi.org/10.1016/S1071-5819(03)00053-3.
Kar, Anuradha, and Peter Corcoran. 2017. “A Review and Analysis of Eye-Gaze Estimation Systems, Algorithms and Performance Evaluation Methods in Consumer Platforms.” *ArXiv*, 16495–519.
Kjeldskov, Jesper, Mikael B. Skov, and Jan Stage. 2004. “Instant Data Analysis: Conducting Usability Evaluations in a Day.” *ACM International Conference Proceeding Series* 82: 233–40. https://doi.org/10.1145/1028014.1028050.
Kolakowska, Agata, Agnieszka Landowska, Mariusz Szwoch, Wioleta Szwoch, and Michal

R. Wrobel. 2013. “Emotion Recognition and Its Application in Software Engineering.” In *2013 6th International Conference on Human System Interactions (HSI)*, 31:532–39. IEEE. https://doi.org/10.1109/HSI.2013.6577877.

Krafka, Kyle, Aditya Khosla, and Petr Kellnhofer. n.d. “Eye Tracking for Everyone.”

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. 2012. “ImageNet Classification with Deep Convolutional Neural Networks.” In *NIPS’12 Proceedings of the 25th International Conference on Neural Information Processing Systems*, 1:1097–1105. https://doi.org/10.2165/00129785-200404040-00005.

Kumar, Ashu, Amandeep Kaur, and Munish Kumar. 2019. “Face Detection Techniques: A Review.” *Artificial Intelligence Review* 52 (2): 927–48. https://doi.org/10.1007/s10462-018-9650-2.

Landowska, Agnieszka. 2013. “Affective Computing and Affective Learning – Methods , Tools and Prospects,” no. June 2013.

———. 2015. “Towards Emotion Acquisition in IT Usability Evaluation Context.” *ACM International Conference Proceeding Series* 29-30-Jun-. https://doi.org/10.1145/2814464.2814470.

LeCun, Yann, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. “Gradient-Based Learning Applied to Document Recognition.” *Proceedings of the IEEE* 86 (11): 2278–2323. https://doi.org/10.1109/5.726791.

Lee, Haeinn, and Ssanghee Seo. 2010. “A Comparison and Analysis of Usability Methods for Web Evaluation : The Relationship Between Typical Usability Test and Bio-Signals Characteristics (EEG , ECG).” *In: The Proceedings of the 2010 DRS Conference*.

Lew, Philip, Luis Olsina, Pablo Becker, and Li Zhang. 2012. “An Integrated Strategy to Systematically Understand and Manage Quality in Use for Web Applications.” *Requirements Engineering* 17 (4): 299–330. https://doi.org/10.1007/s00766-011-0128-x.

Lucey, Patrick, Jeffrey F. Cohn, Takeo Kanade, Jason Saragih, Zara Ambadar, and Iain Matthews. 2010. “The Extended Cohn-Kanade Dataset (CK+): A Complete Dataset for Action Unit and Emotion-Specified Expression.” *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops, CVPRW 2010*, no. July: 94–101. https://doi.org/10.1109/CVPRW.2010.5543262.

Moen, Jin. 2007. “From Hand-Held to Body-Worn: Embodied Experiences of the Design and Use of a Wearable Movement-Based Interaction Concept.” *TEI’07: First International Conference on Tangible and Embedded Interaction*, 251–58. https://doi.org/10.1145/1226969.1227021.

Mollahosseini, Ali, Behzad Hasani, and Mohammad H. Mahoor. 2017. “AffectNet: A Database for Facial Expression, Valence, and Arousal Computing in the Wild.” *IEEE Transactions on Affective Computing*, August, 1–18. https://doi.org/10.1109/TAFFC.2017.2740923.

Nayak, Nandini, and Debbie Mrazek. 1994. “Analyzing and Communicating Usability Data: Now That You Have the Data What Do You Do?” *Conference on Human Factors in Computing Systems - Proceedings* 1994-April (1): 468. https://doi.org/10.1145/259963.260498.

Nielsen, Jakob. 1994. “Guerrilla HCI: Using Discount Usability Engineering to Penetrate the Intimidation Barrier.” *Cost-Justifying Usability*, 245–72. http://www.nngroup.com/articles/guerrilla-hci/.

Nielsen, Jakob, and Rolf Molich. 1990. “Heuristic Evaluation of User Interfaces.” *Conference on Human Factors in Computing Systems - Proceedings*, no. April: 249–56. https://doi.org/10.1145/97243.97281.

Picard, Rosalind W., and Jonathan Klein. 2002. “Computers That Recognise and Respond to User Emotion: Theoretical and Practical Implications.” *Interacting with Computers* 14 (2): 141–69. https://doi.org/10.1016/S0953-5438(01)00055-8.

Rodden, Kerry, Hilary Hutchinson, and Xin Fu. 2010. “Measuring the User Experience on a Large Scale: User-Centered Metrics for Web Applications.” *Conference on Human Factors in Computing Systems - Proceedings* 4: 2395–98. https://doi.org/10.1145/1753326.1753687.

Rowley, Henry A. 1996. “Neural Network-Based Face Detection.”

Russell, James A. 1980. “A Circumplex Model of Affect.” *Journal of Personality and Social Psychology* 39 (6): 1161–78. https://doi.org/10.1037/h0077714.

Scholtz, Jean. 2001. “Adaptation of Traditional Usability Testing Methods for Remote Testing.” *Proceedings of the Hawaii International Conference on System Sciences* 00 (c): 134. https://doi.org/10.1109/HICSS.2001.926546.

Shah, Ruchit, and Yi Yang. 2015. “Going Deeper with Convolutions Christian.” *Population Health Management* 18 (3): 186–91. https://doi.org/10.1089/pop.2014.0089.

Silva Franco, Roberto Yuri da, Rodrigo Santos do Amor Divino Lima, Rafael do Monte Paixão, Carlos Gustavo Resque dos Santos, and Bianchi Serique Meiguins. 2019. “UXmood-A Sentiment Analysis and Information Visualization Tool to Support the Evaluation of Usability and User Experience.” *Information (Switzerland)* 10 (12). https://doi.org/10.3390/info10120366.

Zhang, Huaxun, Yannan Xie, and Cao Xu. 2011. “A Classifier Training Method for Face Detection Based on AdaBoost.” *Proceedings 2011 International Conference on Transportation, Mechanical, and Electrical Engineering, TMEE 2011*, 731–34. https://doi.org/10.1109/TMEE.2011.6199306.

Zimmermann, Philippe G, and Patrick Gomez. 2006. “Extending Usability : Putting Affect into the User-Experience,” no. 1.

Garrett, Jesse James. 2010. *The elements of user experience: user-centered design for the web and beyond.* Pearson Education.

Kaufmann, Morgan. 2016. *Sauro, Jeff, and James R. Lewis. Quantifying the user experience: Practical statistics for user research.* Accessed 11 3, 2020. https://research.utwente.nl/en/publications/review-quantifying-the-user-experience-by-j-sauro-and-j-lewis.

Albert, William, and Thomas Tullis. 2013. *Measuring the user experience: collecting, analyzing, and presenting usability metrics.*

A.J. Bernheim Brush, Morgan Ames, and Janet Davis. 2004. “A Comparison of Synchronous Remote and Local Usability Studies for an Expert Interface.”

Human Factors in Computing Systems. New York: Association for Computing Machinery.

Services, U.S. Dept. of Health and Human. 2006. "The Research-Based Web Design & Usability Guidelines, Enlarged/Expanded edition." Washington.

Scholtz, Jean. 2001. "Adaptation of traditional usability testing methods for remote testing." *International Conference on System Sciences.* Hawaii: IEEE. 8-pp.

Ambler, Scott W. 2004. *The Object Primer: Agile Model-Driven Development with UML 2.0.* Cambridge University Press.

Kling, Rob. 1977. "The organizational context of user-centered software designs." *MIS quarterly* 41-52.

Law, Effie Lai-Chong, Virpi Roto, Marc Hassenzahl, Arnold POS Vermeeren, and Joke Kort. 2009. "Understanding, scoping and defining user experience: a survey approach." *Human factors in computing systems.* Springer.

Bevana, N., Kirakowskib, J., & Maissela, J. 1991. "What is usability?" *Proceedings of the 4th International Conference on HCI.* Stuttgart.

Dumas, Joseph S., Joseph S. Dumas, and Janice Redish. 1999. *A practical guide to usability testing.* Intellect books.

Nielsen, J. 1994. *Usability Engineering.* San Diego: Morgan Kaufmann.

Standardization, International Organization for. 2018. *ISO 9241-11: 2018, Ergonomics of human-system interaction-Part 11: Usability: Definitions and concepts.*

Norman, Don, Jim Miller, and Austin Henderson. 1995. "What You See, Some of What's in the Future, And How We Go About Doing It: HI at Apple Computer." *Conference companion on Human factors in computing systems.* p.155.

Hassenzahl, Marc, and Noam Tractinsky. 2006. "User experience-a research agenda." *Behaviour & information technology* . 91-97.

William S. Green, Patrick W. Jordan. 2002. *Pleasure with products: beyond usability.* London: CRC press.

Gardner, Jessica. 2007. "Remote web site usability testing-Benefits over traditional methods." *International Journal of Public Information Systems.*

CAROL, M. BARNUM. 2020. *USABILITY TESTING ESSENTIALS: Ready, Set... test!* MORGAN KAUFMANN PUBLISHER.

Frøkjær, Erik, Morten Hertzum, and Kasper Hornbæk. 2000. "Measuring usability: are effectiveness, efficiency, and satisfaction really correlated?" *SIGCHI conference on Human Factors in Computing Systems.*

Hornbæk, Kasper, and Effie Lai-Chong Law. 2007. " Meta-analysis of correlations among usability measures ." *SIGCHI conference on Human factors in computing systems.*

Lazar, Jonathan. 2001. *User-centered Web development.* Jones and Barlett Learning.
Anselm Strauss, Juliet Corbin. 1997. *Grouned Theory in Practice.* Sage.
Omodei, Mary, Jim McLennan, Alexander Wearing. 2002. “Head-mounted video cued recall: a methodology for detecting, understanding, and minimising error in the control of complex systems.” Human Decision Making and Control.
Picard, Rosalind W. 1995. *Affective computing.* MIT.