

T.R.
GEBZE TECHNICAL UNIVERSITY
GRADUATE SCHOOL

USING AUGMENTED REALITY INTERFACES FOR
ARTIFICIAL INTELLIGENCE SYSTEMS



BÜŞRA ÖZTÜRK

A THESIS OF MASTER OF SCIENCE
DEPARTMENT OF COMPUTER ENGINEERING

ADVISOR: ASST. PROF. DR. YAKUP GENÇ

MARCH 2024

T.R.
GEBZE TECHNICAL UNIVERSITY
GRADUATE SCHOOL

USING AUGMENTED REALITY INTERFACES FOR
ARTIFICIAL INTELLIGENCE SYSTEMS

BÜŞRA ÖZTÜRK

A THESIS OF MASTER OF SCIENCE
DEPARTMENT OF COMPUTER ENGINEERING

ADVISOR: ASST. PROF. DR. YAKUP GENÇ

MARCH 2024

T.C.
GEBZE TEKNİK ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

YAPAY ZEKA SİSTEMLERİ İÇİN ARTIRILMIŞ
GERÇEKLİK ARAYÜZLERİNİN KULLANILMASI

BÜŞRA ÖZTÜRK

YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

DANIŞMAN: DR. ÖĞRETİM ÜYESİ YAKUP GENÇ

MART 2024



MASTER of SCIENCE JURY APPROVAL FORM

A thesis submitted by Büşra ÖZTÜRK, defended on 07/03/2024 before the jury formed with the 12/02/2024 date and 2024/13 numbered decision of the GTU Graduate Administration Board, has been accepted as a MASTER of SCIENCE thesis in the Department of Computer Engineering.

JURY

MEMBER

(THESIS ADVISOR) : Dr. Öğr. Üyesi Yakup GENÇ

MEMBER

: Doç. Dr. Mehmet GÖKTÜRK

MEMBER

: Dr. Öğr. Üyesi Övgü ÖZTÜRK ERGÜN

APPROVAL

Gebze Technical University Graduate Administration Board
...../...../..... date and/..... numbered decision.

SIGNATURE/SEAL

ABSTRACT

In this study, a new way of visualization is introduced using augmented reality interfaces in a 3D environment for faster understanding and examination of the stages of complex deep learning models. Unlike two-dimensional visualization methods traditionally based on deep learning models, the aim of this approach is to offer users an additional dimension. In this context, in order to examine deep learning models, the advantages offered by 3D augmented reality interfaces and their effects on user experience were evaluated. By taking user opinions, it was decided which visualizations would be useful in artificial intelligence systems. At this stage, the final application was created by processing user experience processes with a design-oriented thinking approach. As the deep learning models progress through the layers in the prediction process, it is displayed how they reach the classification result of an image by examining which stages they go through and which feature maps they produce, which input they focus on, and which input is effective on the output. For this purpose, feature map extraction was first applied on Lenet and Alexnet models, which are CNN models. The application was tested with 51 users in a 2D environment as a desktop application on a computer, and in a 3D environment as an AR application on Microsoft HoloLens-2 glasses. Regarding the traceability of deep learning models, a usability analysis of interfaces in 2D environment and augmented reality interfaces in 3D environment was conducted. First of all, by applying A/B testing, users were divided into two groups, control and experimental groups, and each group was given a single system experience. Usability was measured with a technique called the SUM (Single Usability Metric) model, which includes completion or non-completion, completion times and satisfaction. Users were given tasks and observations were made during and after the task. Satisfaction was calculated with ASQ, which is a user satisfaction measurement survey. Afterwards, the other system they did not use was also tested and comparative comments were made. In this study, the usability of augmented reality interfaces was measured as 90%. It has been concluded that this technology, developed according to the answers received from users, will provide significant benefit in complex deep learning models. It was decided that these systems would contribute to people's use of artificial intelligence systems more effectively.

Keywords: Augmented Reality Interfaces, Deep Learning Models Visualization, User Experience, Human-Computer Interaction.

ÖZET

Bu çalışmada, karmaşık derin öğrenme modellerinin aşamalarının daha hızlı anlaşılması ve incelenmesi için 3 boyutlu ortamda artırılmış gerçeklik arayüzleri kullanılarak yeni bir görselleştirme yolu tanıtılmıştır. Geleneksel olarak derin öğrenme modelleri üzerinde yapılan iki boyutlu görselleştirme metotlarından farklı olarak bu yaklaşımın amacı, kullanıcılara ek bir boyut sunmaktır. Bu bağlamda derin öğrenme modellerinin incelenmesi amacıyla 3 boyutlu artırılmış gerçeklik arayüzlerinin sunduğu avantajlar ve kullanıcı deneyimine etkileri değerlendirilmiştir. Kullanıcı görüşleri alınarak yapay zeka sistemlerinde hangi konuda görselleştirmenin faydalı olabileceğine karar verilmiştir. Bu aşamada tasarım odaklı düşünme yaklaşımı ile kullanıcı deneyimi süreçleri işlenerek nihai uygulama oluşturulmuştur. Derin öğrenme modellerinin tahmin etme sürecinde katmanlarda ilerlerken hangi aşamalardan geçip hangi öznelilik haritalarını çıkardığı, hangi girdiye odaklanıp, hangi girdinin çıktı üzerinde etkili olduğunu inceleyerek bir görüntünün sınıflandırma sonucuna nasıl ulaştığı görüntülenmiştir. Bunun için öncelikle öznelilik harita çıkarımı, birer CNN modeli olan Lenet ve Alexnet modelleri üzerinde uygulanmıştır. Uygulama 2 boyutlu ortam olarak bilgisayarda masaüstü uygulaması, 3 boyutlu ortamda da Microsoft Hololens-2 gözlüğünde AR uygulaması olarak 51 kullanıcıyla test edilmiştir. Derin öğrenme modellerinin izlenebilirliği konusunda 2 boyutlu ortamdaki arayüzler ile 3 boyutlu ortamda artırılmış gerçeklik arayüzlerinin kullanılabilirlik analizi yapılmıştır. Öncelikle A/B testi uygulanarak kullanıcılar kontrol ve deney grubu olarak iki gruba ayrılmış ve her bir gruba tek bir sistem deneyimletilmiştir. Kullanılabilirlik, SUM (Single Usability Metric) modeli adı verilen tamamlanma veya tamamlanmama durumlarını, tamamlanma sürelerini ve memnuniyeti içeren bir teknikle ölçülmüştür. Kullanıcılara görevler verilmiş ve görev sırasında ve sonrasında gözlem yapılmıştır. Memnuniyet, kullanıcı memnuniyeti ölçüm anketi olan ASQ (After Scenerio Questionerie) ile hesaplanmıştır. Sonrasında kullanmadıkları diğer sistem de denetilerak karşılaştırmalı yorumlar alınmıştır. Bu çalışmada artırılmış gerçeklik arayüzlerinin kullanılabilirliği %90 olarak ölçülmüştür. Kullanıcılardan alınan cevaplara göre gelişen bu teknolojinin karmaşık derin öğrenme modellerinde kayda değer bir fayda sağlayacağı sonucu elde edilmiştir. Bu sistemlerin insanların yapay zeka sistemlerini daha etkin bir şekilde kullanmalarına katkısı olacağına karar verilmiştir.

Anahtar Kelimeler: Artırılmış Gerçeklik Arayüzleri, Derin Öğrenme Modellerinin Görselleştirilmesi, Kullanıcı Deneyimi, İnsan-Bilgisayar Etkileşimi.

ACKNOWLEDGEMENTS

I would like to express my gratitude to my advisor, Asst. Prof. Dr. Yakup GENÇ, who provided unwavering support throughout my master's education and guided me with his knowledge and experience. I am also grateful to Musa ALTILI, the Team Leader of Augmented and Virtual Reality at Havelsan Inc., who accompanied me throughout my study and provided assistance with his ideas. Therefore, I extend my thanks to Havelsan Inc. for their contribution, especially for providing the mixed reality glasses. Lastly, I would like to thank my husband, Orhan ÖZTÜRK, for their support.



TABLE OF CONTENTS

	<u>Page</u>
ABSTRACT	v
ÖZET	vi
ACKNOWLEDGEMENTS	vii
TABLE OF CONTENTS	viii
LIST OF SYMBOLS AND ABBREVIATIONS	x
LIST OF FIGURES	xi
LIST OF TABLES	xiv
1. INTRODUCTION	1
1.1. Purpose of the Study	3
1.2. The Study Methodology	4
2. LITERATURE REVIEW	5
2.1. Perception and Visual Perception in Humans	5
2.2. The Importance of UX for Human Perception	6
2.3. Augmented Reality	14
2.3.1. Usability Metrics	15
2.3.2. Interaction Techniques in AR	19
2.3.3. AR Display Types	20
2.4. Deep Learning	21
2.5. AI and AR Applications	23
2.6. Traceability of DL Models	23
2.6.1. Visualizing Models with 2D Interfaces	25
2.6.2. Visualization of Models with 3D Interfaces	34
3. METHODOLOGY	38
3.1. DL Model Creation	38
3.2. DL Models Visualization Application	45
3.2.1. Visualization in 2D Environment	45
3.2.2. Visualization with AR Interfaces in 3D Environment	49
3.3. UX Study	56
3.3.1. Empathy and Problem Identification	56
3.3.2. Idea Development and Prototype Creation	60
3.3.3. User Testing	61
3.3.4. Determine Usability Metric	64
4. EXPERIMENT AND RESULTS	68
4.1. Creating a User Group	68
4.2. Experiment	68
4.3. Data Collection	70
4.4. Results	72
4.4.1. Satisfaction Measurement	73
4.4.2. Task Completion	74
4.4.3. Time Measurement	75

5. CONCLUSIONS	83
REFERENCES	85
BIOGRAPHY	90
PUBLICATIONS AND PRESENTATIONS FROM THE THESIS	91



LIST OF SYMBOLS AND ABBREVIATIONS

GTU	:	Gebze Technical University
AR	:	Augmented Reality
DL	:	Deep Learning
AI	:	Artificial Intelligence
UX	:	User Experience
UI	:	User Interfaces
MRTK	:	Mixed Reality Toolkit
ASQ	:	After-Scenario Questionnaire
SUM	:	Single Usability Metric
ISO	:	International Standards Organization
MLOps	:	Machine Learning Operations
AHAT	:	Articulated Hand Tracking



LIST OF FIGURES

	<u>Page</u>
Figure 1.1: Plan for this study.	4
Figure 2.1: Design thinking is an approach that includes a cyclical process [14].	10
Figure 2.2: Usability guidelines by ISO, Nielsen and Shneiderman [13].	11
Figure 2.3: The ASQ questions evaluate the overall satisfaction of a system [21].	13
Figure 2.4: SUM model is a quantitative measure used to assess the overall usability of a product or system [22].	14
Figure 2.5: Pinhole camera model used in AR systems [28].	17
Figure 2.6: Classification of DL Visualization methods [40].	25
Figure 2.7: Neptune is an open source library that allows users to regularly save the data they create during the model development process [41].	26
Figure 2.8: Wandb is very effective in monitoring and understanding the performance of the model [42].	27
Figure 2.9: Tensorboard is used to monitor the model's performance, detect errors, and help improve the model [43].	28
Figure 2.10: Visdom is a library that enables interactive visualization of data through a browser-based interface [44].	28
Figure 2.11: Hiplot is generally used for hyperparameter optimization and results exploration [45].	29
Figure 2.12: CNNVis is a tool that focuses on analyzing a snapshot of the CNN models [46].	30
Figure 2.13: Neural Networks Playground is a simple tool for NN model exploration and visualization [47].	30
Figure 2.14: Netron provides a simple and intuitive interface for visualizing models in various formats [48].	31
Figure 2.15: ANN Visualizer is a python library for visualization NN [49].	31
Figure 2.16: Visualization with activation maximization method [50].	33
Figure 2.17: Discriminateness of different methods for calculating the saliency map [53].	34
Figure 2.18: Network visualized with VR in Unreal Engine [55].	34
Figure 2.19: Deep Visualization Toolbox is used to create neuron-by-neuron visualizations using regular optimization [56].	35
Figure 2.20: Steps of the description process from the user's perspective in virtual space and model architecture. [57]	35
Figure 2.21: DeepVisionVR application allowed the user to freely navigate the layered network, retrieve and place images, move and scale layers for better readability, perform feature visualization, and export the results [58].	36
Figure 2.22: This visualization technique offers exciting possibilities for understanding and interacting with complex models in immersive environments [59].	37
Figure 2.23: It is a VR visualization tool specifically targeted at individuals with limited computational science knowledge to perform deep learning tasks [60].	37

Figure 3.1: Parts and flow diagram of the study.	38
Figure 3.2: Lenet architecture [61].	39
Figure 3.3: Architecture of the lenet model used in this study.	41
Figure 3.4: Values of the filters in convolution layers recorded for visualization in this study.	41
Figure 3.5: Filters of the model used in the study are saved as image files.	42
Figure 3.6: Feature maps obtained from the test data of the model used in the study are saved as image files.	42
Figure 3.7: Architecture of the alexnet model used in this study.	44
Figure 3.8: Filters of the alexnet model used in the study are saved as image files.	44
Figure 3.9: Giving the cat image to the model, extracting the feature maps and the result.	45
Figure 3.10: Created image objects for feature maps on layers for the Lenet Model.	46
Figure 3.11: Menu and screens that open when selection is made.	47
Figure 3.12: Layers opened after selecting image "0".	47
Figure 3.13: Layers opened after selecting the cat image.	48
Figure 3.14: The content of the layers after clicking the buttons.	48
Figure 3.15: The content of the layers after clicking the buttons.	48
Figure 3.16: Viewing feature maps selected from layers.	49
Figure 3.17: The hand tracking system uses a people's palms and fingers as input [64].	50
Figure 3.18: Hololens-2 sensors [64].	50
Figure 3.19: Hand menu that allows switching between models.	53
Figure 3.20: Formation of the selected model in front of the wall.	53
Figure 3.21: Expressing the volumes of layers with cubes.	54
Figure 3.22: Examination of layers in the Lenet model.	54
Figure 3.23: Examination of layers in the Alexnet model.	54
Figure 3.24: Close examination of images focusing on similar features.	55
Figure 3.25: Object manipulation techniques used in the study.	55
Figure 3.26: Stages where visualization can be used in the DL system.	56
Figure 3.27: User personas are user profiles that represent the wants and needs of the target audience.	59
Figure 3.28: Personas created for the study.	59
Figure 3.29: Creating a Wireframe for the prototype of a 2D application.	60
Figure 3.30: Application created after prototype.	61
Figure 3.31: The survey was accessed via QR code.	65
Figure 3.32: The survey used in the study.	65
Figure 4.1: Years of user's experience.	70
Figure 4.2: Users' interests.	70
Figure 4.3: Testing 2d system with users.	71
Figure 4.4: Testing AR system with users.	71
Figure 4.5: Classification of user answers.	72
Figure 4.6: SUM usage pattern.	73
Figure 4.7: Averages of answers for ASQ and its equivalent for 5.	74
Figure 4.8: Task completion status.	74
Figure 4.9: Duration of tasks (s).	75
Figure 4.10: SUM score calculation based on user answers for Task-1.	76
Figure 4.11: SUM score calculation based on user answers for Task-2.	76

Figure 4.12: SUM score calculation based on user answers for Task-3.	77
Figure 4.13: SUM score calculation based on user answers for Task-4.	77
Figure 4.14: SUM score calculation based on user answers for Task-1.	78
Figure 4.15: SUM score calculation based on user answers for Task-2.	78
Figure 4.16: SUM score calculation based on user answers for Task-3.	79
Figure 4.17: SUM score calculation based on user answers for Task-4.	79
Figure 4.18: Comparison of system 1 and 2.	80
Figure 4.19: Comparison of system 1 and 2.	80
Figure 4.20: User answers to 2D interface questions.	82
Figure 4.21: User answers to augmented reality questions.	82



LIST OF TABLES

	<u>Page</u>
Table 3.1: The content of the object created for layers in Unity scene.	46
Table 3.2: Questions asked for user needs analysis.	57
Table 3.3: User answers for analyzing user needs.	58
Table 3.4: User interests.	62
Table 3.5: User experiences.	62
Table 3.6: Questions asked after user testing.	63
Table 3.7: Tasks for 2D Interfaces on laptop application.	66
Table 3.8: Tasks for AR Interfaces on hololens-2 application.	67
Table 4.1: Occupational distribution of users.	69
Table 4.2: Educational status distribution of users.	69

1. INTRODUCTION

Perception is a term that refers to the process by which a person understands the world around them through their senses. The perception process generally occurs through the five senses of sight, hearing, smell, taste and touch. Perception includes not only the collection of information coming through the senses, but also the interpretation, meaning and organization of this information. It also includes the individual's experiences, prior knowledge, cultural influences, and other factors. That is, the perceiver not only reacts to stimuli from the external world, but also derives meaning by combining these stimuli with his own previous experiences and understanding. Visualization is, for some, the most important way of perception. Visualization, the process of presenting or creating visual information, has a significant impact on perception. According to Guzman, visualization is not just about what is perceived with the eyes. It is part of a vision that also includes psychological perceptions. Guzman divides visualization into four classes: isomorphic, homomorphic, analogical, and diagrammatic [1].

User interfaces enable people to interact and communicate with computers. For the design of these interfaces and comfortable use by users, it is necessary to understand the users, master their experiences and plan the overall functionality of the system.

The evolution of machine learning reflects a process in which interest in artificial intelligence continues to increase from past to present. This increasing interest has led to the emergence of deep learning architectures, which are one of the most preferred artificial intelligence algorithms today. Deep learning architectures have become one of the important tools that provide solutions to various artificial intelligence problems. In this context, it has the capacity to produce intelligent solutions in many fields such as industry, medicine, robotics, image processing, computer vision, object detection, voice processing-recognition, translation, future prediction and finance [2]. DL models are built based on artificial neural networks. These neural networks learn the input data along with the weight values determined during the training process and contain hidden layers. Once training is complete, the model makes a prediction using these weights given a new input. In this process, the weights in the hidden layers are constantly updated by the feedback method, which

enables the model to make better predictions and increase its performance. It is important to evaluate whether the model is working effectively, to understand where it succeeds and where it makes mistakes. These evaluations are often made through a variety of methods and criteria. Visualization is a prominent tool in this evaluation process. To better understand the performance of the model and to enable data scientists and users to better understand the behavior of the model, visualization of various metrics and results is of great importance. These visual tools reveal the model's strengths and weaknesses more clearly, making it easier to understand the complexities within the model.

User interfaces can generally be in two-dimensional environments, but it is also possible to provide an experience in a three-dimensional environment by adding one more dimension. This approach aims to improve people's cognitive, perceptual and motor skills as they interact with the physical world. Particularly in the context of augmented reality, it is important to create a three-dimensional representation of the world so that digital objects can coexist with physical objects. AR interfaces combine parts of the real world with synthetic, computer-generated images, enriching the user's perception with additional information [3]. In this way, users can establish a deeper and more interactive connection between the real and virtual worlds.

Visualization plays a major role in the implementation of deep learning technologies, which is a subset of artificial intelligence, in order to facilitate users in the data collection, training, installation and usage stages. Visualization is an important tool at a number of stages in the application development process, from tracking data and models to providing an effective environment for users through AR interfaces in the final application. During the application development phase, the status of data and models can be monitored instantly thanks to visualization tools. This gives developers better understanding and feedback, allowing them to make rapid improvements. At the same time, the use of visualization in recent applications such as augmented reality interfaces contributes to users having an interactive and more meaningful experience. In this study, visualization was provided on deep learning models using augmented reality interfaces. Two versions of the same application were created to be tested in a 2D computer environment and Microsoft Hololens-2 glasses, and user testing was conducted with 51 users with different experience groups.

1.1. Purpose of the Study

Deep learning models often work with complex and high-dimensional data structures. Understanding and following the inner workings of these models can be difficult for developers and users. Users can interact more effectively through visual tools that enable them to better understand complex data and algorithms. Nowadays, there are many studies on visualization of deep learning models. These studies focused on issues such as model understandability, interpretability, visualization techniques and in-model activations. Visualization techniques have been developed that reveal input stimuli that activate individual feature maps at any layer of many deep learning models. These techniques are used to understand the learning process and its activations. AR interfaces can make the way deep learning models work more understandable and tractable. For example, visually seeing how the model responds to certain features or layers can help users better understand the model's behavior, enabling understanding of the model's inferences, recommendations, or classifications. The aim of this study is to investigate user experience with augmented reality interfaces, which are thought to play a more effective role in user perception when visualizing deep learning models.

The topics contributed to the literature by this study can be listed as follows:

- Visualization of deep learning models was enabled on the Hololens device, which is used for augmented reality interfaces that enable seeing virtual objects in the real environment,
- By measuring users' perceptions of the 2D interfaces on the computer and the interfaces in the 3D environment containing augmented reality on the Hololens device, the usability of this system was measured by going through user experience stages.

Within the scope of this thesis, firstly, in the second chapter, the literature is reviewed in detail and the traceability of user experience, augmented reality, deep learning and deep learning models are investigated. In the third chapter, the methodology of the study, from model creation to visualization, and the user experience study are explained. In the fourth section, the results obtained from the user tests were examined. The last chapter, the fifth, contains a summary of the results and analysis of the study.

1.2. The Study Methodology

In this section, the work plan and steps are mentioned. The study methodology shown in **Figure 1.1** is a working step.

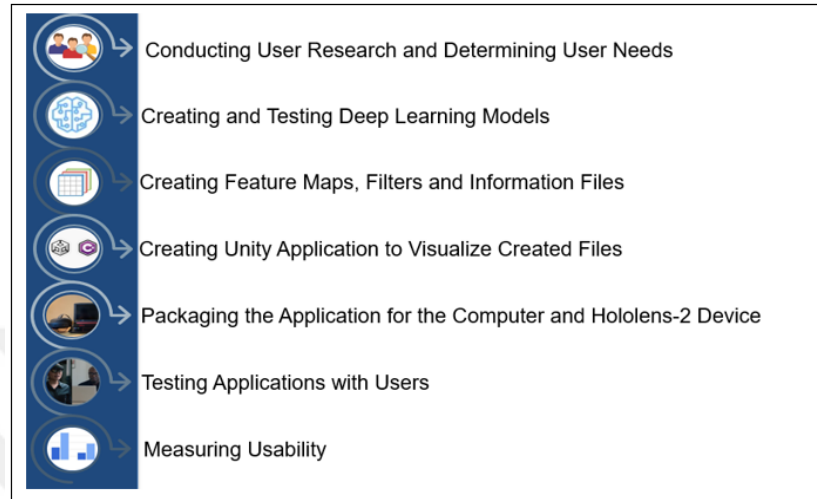


Figure 1.1: Plan for this study.

In the study, first of all, two selected deep learning models (Lenet and Alexnet) were created and feature maps were obtained by testing them with test data. These image files, created using Python language and Jupyter notebook, were exported to the Unity game engine. An application was created to visualize image files received with Unity. This application was first made with libraries that could run on computers. It was later revised with libraries (MRSDK) that can run on the Hololens device. A package was created for two different uses and turned into a workable application. For user testing, statistical methods in this field were first examined. The appropriate metric was found. Survey questions were prepared. Users were made to experience the system individually within the scope of A/B testing. In this process, tasks were given to the user. The duration of these tasks, the user's satisfaction and whether they were able to complete the task were noted. Survey questions were answered. Accordingly, satisfaction was measured. User suggestions were also taken. According to these results, the usability of the system was measured according to the determined metric. Analyses were made.

2. LITERATURE REVIEW

Traceability and interpretability of complex deep learning models is an important issue. It may be necessary to provide reliable explanations to users, especially in critical applications (such as health, finance, security) where these models are used. Monitoring and examining deep learning models is important to understand the internal structure and behavior of these models, to explain why the model makes certain decisions, and to improve the reliability of the model overall. Efforts to track complex deep learning models in a way that is compatible with human perception are often focused on visualization. There are many studies regarding this.

Under this section, in section 2.1, research on people's visual perception is first mentioned. Section 2.2 explains the importance of user experience in people's perception. Augmented reality is mentioned in chapter 2.3. Deep learning is mentioned in section 2.4. In section 2.5, the issue of traceability of deep learning models is discussed.

2.1. Perception and Visual Perception in Human

Perception is a complex and multifaceted concept that may differ between individuals. People perceive the events around them through their sense organs. There are five basic sense organs: sight, hearing, touch, smell and taste. These sensory organs perceive various stimuli and convert these stimuli into neural signals and transmit them to the brain. The brain interprets environmental information by processing these neural signals. Theoretical approaches to perception may include various perspectives to understand how the individual processes, interprets and gives meaning to sensory data. The perceptual process relies not only on sensory input, but also on a variety of factors, such as the individual's cognitive processing, previous experiences, and cultural context. Therefore, the way we perceive something is shaped by sensory data as well as our personal history and other cognitive factors [4].

Visual perception, a type of perception, includes the perception of objects, colors, patterns, distance and size, as well as visual elements such as depth and perspective. Visual perception is a complex process and people use previous experiences and

learned predictions to make sense of the information they receive through the eyes. When people process physical information from their environment, they interpret this information based on past experiences and learned knowledge. Perception does not directly depend on the stimulus; on the contrary, correct perception can occur if the relationship between the stimuli is understood and there are sufficient stimuli. Environmental objects, situations and spatial arrangement must be sufficient and clear stimuli to be perceived correctly [5]. In a design, information about the relationship of design elements with each other and with the viewer, as well as how the design is interpreted in the perception process, is of vital importance for designers. Studies in this field are discussed in gestalt principles, which are a form of psychology studies [6]. Gestalt psychology has a perspective that argues that a whole is more than its parts. How design elements are arranged, grouped and put together affects the viewer's perception. This involves an approach to understanding how the viewer perceives visual elements and creates holistic meaning. Using these principles, designers work to understand the viewer's eye and mind coordination, organize design elements and ensure effective communication. According to Koffka [7], the theory of language and symbolic expressions stands out as a central concept for understanding human thought and imagination. Accordingly, language, symbols and other symbolic expressions are the building blocks of thought, and through these symbols, people try to understand the world, organize thought processes and develop imagination.

2.2. The Importance of UX for Human Perception

UX refers to the overall experience a person has when using a product, service or system. Don Norman popularized the term user experience, stating that it encompasses all aspects of end users' interaction with the company, its services, and its products [8]. This interaction includes factors such as the emotions the user feels during this process, the level of satisfaction, ease or difficulty, efficiency and satisfaction. A good user experience makes it easier for the user to understand the product. The increase in understanding makes the user's interaction with the product more effective and efficient. Human perception is a critical factor in understanding, evaluating, and improving this experience. In experience design processes, methods

such as user research, feedback analysis and user testing are important tools to understand human perception and improve user experience.

- UX Research and Design Thinking

User experience, or user research, is defined as the study of people's behaviors, motivations, and needs in a specific context, which affects how people understand and use things in their daily lives and may include how these factors may change over time. UX research methods often overlap with other frameworks such as user (or human)- centered design, user design, and implementation science. These approaches often work in parallel and ultimately inform each other to arrive at the final solution (i.e. a product, process, or experience). UX research methods can be used to obtain feedback on the design, how well it achieves results, and offer suggestions for revisions. User Experience (UX) research is an important part of the design process and focuses on understanding users, their behaviors, needs and preferences. The goal is to create products, services or systems that are not only usable but also enjoyable and effective for the target audience. UX research involves a variety of methods and techniques for collecting information about users and their interactions with products or services. Key elements of UX research:

1. User Personas: Creating user personas involves developing fictional characters that represent different types of users. These personas help designers and developers understand the various needs, goals, and characteristics of their target audience.
2. User Interviews: Conducting one-on-one interviews helps gather in-depth qualitative insights from users. This method allows researchers to explore users' thoughts, feelings, and experiences regarding a product or service.
3. Surveys and Questionnaires: Surveys are useful for collecting quantitative data and opinions from a larger group of users. Surveys can provide valuable insights to understand user demographics, preferences, and satisfaction levels.
4. Usability Testing: This involves observing users as they interact with a prototype or finished product. Usability tests help identify issues with navigation, functionality, and overall user experience.

5. Card Sorting: Card sorting involves users organizing information into categories. This helps designers understand how users mentally categorize and prioritize information.
6. Context Study: Researchers use this method to understand how users interact with products or services in real-world situations by observing them in their natural environments. This method is particularly useful for understanding the context in which a product will be used.
7. Competitive Analysis: Examining competitors' products helps identify industry standards, best practices, and areas where a product can differentiate itself.
8. Analytics: Analyzing user data can provide quantitative insights such as user behavior, popular features, user flow, and areas of high or low engagement.
9. A/B Testing: This involves comparing two versions of a design (A and B) to determine which one performs better in terms of user engagement, conversion rates, or other relevant metrics.
10. Accessibility Testing: Ensuring whether a product is accessible to users with disabilities is important as a critical element of UX research. Accessibility testing aims to create inclusive designs.

Effective UX research involves a combination of these methods, based on the specific goals and constraints of the project. An important part of a successful UX research and design process is constant iteration and user feedback. Collectively, the approach to identifying and addressing some design challenges with innovative solutions can be called “design thinking [9]. Design thinking is a methodology used to solve a problem or improve a product. The term “design thinking” was popularized by David Kelley and Tim Brown, who studied design and entrepreneurship at Stanford University. This term began to be used in the early 2000s to describe the methods and principles developed at design firms such as IDEO [10]. This approach aims to find creative and effective solutions by focusing on the needs of the user. Design-oriented thinking stages, as seen in Figure 2.1;

1. Empathy:

- Empathize to understand the needs of your users,

- Making observations, conducting interviews and interacting with users,
- Trying to understand users' real needs and feelings.

2. Define:

- Analyzing collected information and determining user needs,
- Clearly defining the problem area and focusing on understanding the problem.

3. Ideate:

- Generating various solution ideas using creative thinking techniques,
- Encouraging critical thinking and generating a wide range of ideas,
- Sharing ideas using techniques such as group brainstorming.

4. Prototyping (Prototype):

- Creating prototypes based on the best selected ideas, (Prototypes serve as a way to solidify and test ideas.)
- Gain feedback by interacting prototypes with users.

5. Testing and Implementation:

- Testing prototypes with users and collecting feedback,
- Observing users' experiences with the product or solution, Refining the design using feedback and revising prototypes as necessary.

These phases are often a cyclical process and support a design process where feedback is continuously integrated. Design thinking offers a usable and flexible approach to improving problem-solving processes in general, not just product design [11]-[13].

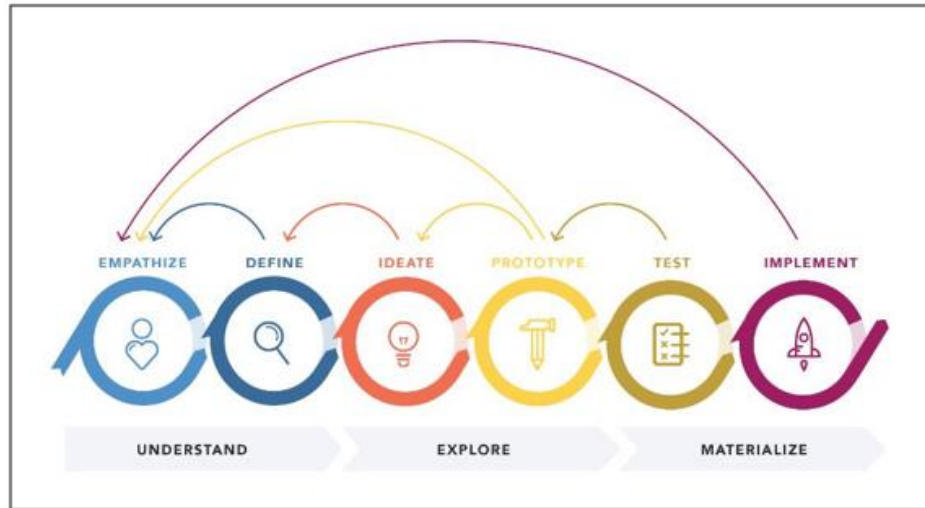


Figure 2.1: Design thinking is an approach that includes a cyclical process [14].

Jakob Nielsen is a renowned usability expert and has made significant contributions especially to user experience. Nielsen emphasizes the principles of simplicity, clarity and user-centered design. He emphasizes the importance of testing with real users, uncovering real-world problems and improving the overall user experience. He says it's important to provide users with clear feedback about system status. Users should always be aware of what is happening in the system, what it is doing, and what has changed. It is important to provide users with the ability to control the system and easily recover from errors. Consistency in design elements and compliance with established standards are critical to usability. Users shouldn't have to think about whether different words, situations, or actions mean the same thing. He suggests that interfaces should be designed that minimize cognitive load, allowing users to focus on their tasks. These principles are part of a broader usability framework and are intended to guide designers in creating effective, efficient and satisfying interfaces for users [15].

Usability is generally understood as a qualitative characteristic that evaluates the quality of use or how easy the application is to use. ISO offers a set of usability guides; Nielsen and Shneiderman are the most widely accepted, followed, and cited guidelines in the specialty literature. **Figure 2.2** compares specific guidelines advocated by the cited authors [13].

ISO	Nielsen	Shneiderman
Productivity	• Consistency and standards • Error prevention	• Strive for consistency
Sufficiency	• Flexibility and efficiency of use • Aesthetic and minimalist design • Help users recognize, diagnose, and fix errors	• Design dialog for closing • Offer simple error handling • Reduce short-term memory load
Satisfaction	• User control and freedom • Visibility of system status • Match between system and the real world • Recognition rather than recall • Help and documentation	• Offer informative feedback • Enable frequent users to use shortcuts • Permit easy reversal of actions • Support internal locus of control

Figure 2.2: Usability guidelines by ISO, Nielsen and Shneiderman [13].

• Usability Metrics

Usability metrics include various criteria for measuring how usable a product or service is in the eyes of users. By incorporating real user data into the design process and measuring the benefits or harms of each change, it is necessary to fully understand and explain what impact the changes have on user behavior.

Efforts to standardize usability metrics aim to collect data consistently and make them comparable in usability research. These standardization efforts ensure universal acceptance of methods and metrics in the field of usability, helping researchers and designers analyze data more effectively and share their results. Considering the studies aimed at standardizing usability metrics,

1. SMEQ (Single Ease Question): SMEQ is a simple single question measurement tool used to evaluate the usability of a product or service. It usually consists of one easy question posed to the end user to evaluate the user experience. For example, “How easy was it to use this product or service?” A question like this is asked and users evaluate this question on a certain scale (such as 1-5).
2. SEQ (Single Ease Question): SEQ is another single question measurement tool used to evaluate the usability of a product or service. Like SMEQ, the user is asked just one easy question and they rate it on a specific scale. For example, “How easy was it to use this website?” A question like this can be used [16].
3. UME (Usability Magnitude Estimation): UME is a method used to estimate the usability of a product. Users are asked to estimate the usability level of a

product before using it or with very limited experience. Then, after having their actual experience, it is compared to their predictions. This method can be used to obtain preliminary information about the usability of the product or service [17].

4. SUMI (Software Usability Measurement Inventory): SUMI is a widely used standardized tool for measuring software usability. This tool includes more than 50 metrics to evaluate users' experience using the software. SUMI measures how usable users find the software overall and uses structured surveys to determine how users rate specific features [18].
5. SUS (System Usability Scale): SUS is a widely used standardized measurement tool. SUS uses a 10-item survey to evaluate how usable users find a product. This survey evaluates factors such as the product's learning curve, memorability, ease of interaction, and satisfaction.
6. QUIS (Questionnaire for User Interface Satisfaction): QUIS is a standardized measurement tool designed to measure user interface satisfaction. QUIS uses a structured survey to measure how satisfactory users find an interface. This survey evaluates factors such as difficulties, satisfaction, and expectations that users experience during their interaction with the interface [19].
7. PSSUQ (Post-Study System Usability Questionnaire): PSSUQ is a widely used standardized measurement tool to evaluate the usability of a system or product. It uses a structured survey to evaluate how usable users find a product. PSSUQ measures the satisfaction, difficulties, and overall perception of usability that users experience after interacting with the product [20].
8. ASQ (After-Scenario Questionnaire): ASQ is a user experience research tool and is often used as an after-experience survey. This survey is designed to measure a user's opinions about the usability of the product or service after an experience. We can say that ASQ is a suitable tool for usability measurement because it is designed to measure users' experience and satisfaction. The survey allows users to evaluate in detail their experience with the use of the product or service [21].

		1	2	3	4	5	6	7	NA
1. Overall, I am satisfied with the ease of completing the tasks in this scenario ☐	strongly disagree	☐	☐	☐	☐	☐	☐	☐	strongly agree ☐
2. Overall, I am satisfied with the amount of time it took to complete the tasks in this scenario ☐	strongly disagree	☐	☐	☐	☐	☐	☐	☐	strongly agree ☐
3. Overall, I am satisfied with the support information (online-line help, messages, documentation) when completing the tasks ☐	strongly disagree	☐	☐	☐	☐	☐	☐	☐	strongly agree ☐
		1	2	3	4	5	6	7	NA

Figure 2.3: The ASQ questions evaluate the overall satisfaction of a system [21].

These tools are standardized measurement tools commonly used in usability research, and each differs in certain aspects. SUMI measures how generally users find the software usable, while SUS evaluates the overall usability of a product or system. QUIS measures user interface satisfaction, while PSSUQ evaluates experiences with a product's usability. These standardized tools ensure that data is collected consistently and made comparable in usability studies. Using these tools, researchers and designers can more effectively evaluate usability metrics and identify areas they need to focus on to improve their products or systems.

In a summative usability evaluation, a variety of metrics are made available to the analyst to benchmark the usability of a product. There is general agreement across the ANSI 2001 and ISO 9241 standards boards about what the dimensions of usability (effectiveness, efficiency, and satisfaction) are and what metrics are most commonly used to measure these dimensions. It includes measures of effectiveness, completion rates, and errors; productivity is measured by time spent on task, and satisfaction is summarized using any of a number of standard satisfaction surveys (either collected on a task-by-task basis or at the end of a testing session.) In an effort to fully represent the entire construct of usability and create a single measure of usability, SUM (Single Usability Metric) model was created. SUM aims to express usability with a single numerical value and is often presented as a score or scale. Rather than conducting complex research to understand the overall usability of a product, it summarizes users' experiences into a single metric. Therefore, SUM differs from other standardized usability measurement tools because it targets a single numerical value. Other measurement tools often include multiple metrics or dimensions to evaluate the usability experience in more detail [22], [23].

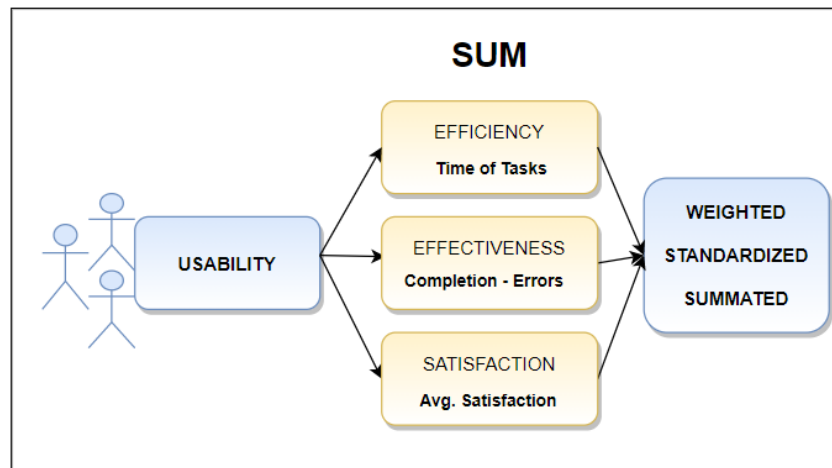


Figure 2.4: SUM model is a quantitative measure used to assess the overall usability of a product or system [22].

Which measurement tool to use will depend on the research objectives, the complexity of the product, and the level of detail of the measurement.

2.3. Augmented Reality

AR can be defined as a newer technological system in which virtual objects are added to the real world in real time during the user's experience [24]. Azuma et al say that an AR system should: (1) combines real and virtual objects in a real environment; (2) works interactively and in real time [25]. Augmented Reality aims to simplify the user's life by bringing virtual information not only to their immediate surroundings but also to any indirect view of the real-world environment, such as live video streaming. AR enhances the user's perception of and interaction with the real world. While Virtual Reality (VR) technology, or Virtual Environment as Milgram calls it, immerses users in a completely synthetic world without seeing the real world, AR technology increases the sense of reality by overlaying virtual objects on top of the real world in real time [26]. In the article published by Zhou, Duh and Billinghurst in 2008, a ten-year review of ISMAR (International Symposium on Mixed and Augmented Reality) was made and the trends in monitoring, interaction and imaging in augmented reality were discussed [27].

2.3.1. Tracking Techniques in AR

Tracking techniques used in Augmented Reality (AR) applications are used to monitor the user's device and/or environment and place virtual objects in harmony with the real world. One of these, the sensor-based tracking method, is used to determine the position and/or orientation of an object or device using data from various sensors [27]. These methods estimate position and orientation in real time by processing data from various sensors (e.g., accelerometers, gyroscopes, magnetometers, laser sensors, etc.). Gyroscopes measure the rotational speed and orientation of an object. This information can be used to detect rotations of the device and calculate its position. Magnetometers measure the direction of the magnetic field and can use this information to determine the position of the device. Magnetometers are especially useful for orientation determination or situations where the device's location must be determined relative to a reference point, such as north. Laser sensors are used to measure distances around an object. These sensors can be used to detect obstacles around the object and determine the position of the device. Laser sensors are especially useful in applications that require precise positioning. Ultrasonic sensors measure distance using the echo time of sound waves. These sensors can be used to detect distances around an object and determine the location of the device. Ultrasonic sensors are often used to track objects at close ranges. These sensor-based tracking methods are often combined or used together to provide more precise and reliable location and orientation estimates.

The other tracking method, vision-based tracking techniques, can use image processing methods to calculate camera pose relative to real-world objects and are thus similar to systems that dynamically correct errors. In Augmented Reality (AR) applications, computer vision techniques play a critical role in understanding and interacting with the real world. It enables virtual objects to be integrated into a real-world scene in real time, allowing the user to receive a composited image occurring from the camera's perspective. Most of the existing tracking techniques in computer vision can be divided into two classes: feature-based and model-based. Feature-Based Tracking; detects salient features in a real-world scene (e.g. corners, edges, color changes, etc.) and determines the object's position and/or orientation by tracking the movement of these features. Feature-based tracking;

- Feature Extraction: Extracting salient features from real-time images,
- Matching: Making a match between the extracted features,
- Positioning: Determining the location of matching features,
- Tracking: It features realtime tracking of the object's location and/or orientation.

This method is to find a correspondence between 2D image features and their 3D world coordinates. Camera pose can be found by projecting the 3D coordinates of features onto the observed 2D image coordinates and minimizing their distance to the corresponding 2D features [27]. Constraints for camera pose estimation are mostly determined using point features. The reconstruction/recognition phase uses the data obtained from the first phase to reconstruct the real-world coordinate system. Assuming a calibrated camera and perspective projection model, if there is a point with coordinates $(x, y, z)^T$ in the camera's coordinate frame, its projection onto the image plane is $(x/z, y/z, 1)^T$.

In point constraints, there are two main coordinate systems, as shown in **Figure 2.5**: the world coordinate system W and the 2D image coordinate system. Let $p_i(x_i, y_i, z_i)^T, i = 1, \dots, n, n \geq 3$ be the set of 3D noncollinear reference points in the world frame coordinate, and let $q_i(x_i', y_i', z_i')^T$ be the camera space coordinates corresponding to; p_i and q_i are related by the following transformation (2.1, 2.2):

$$q_i = R p_i + T \quad (2.1)$$

$$\text{where } R = \begin{pmatrix} r_1^T \\ r_2^T \\ r_3^T \end{pmatrix} \text{ and } T = \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix} \quad (2.2)$$

rotation matrix and a translation vector respectively.

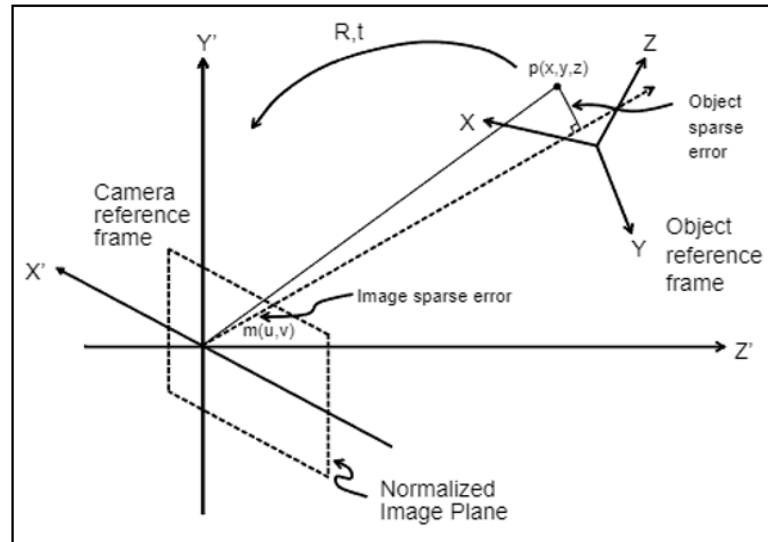


Figure 2.5: Pinhole camera model used in AR systems [28].

Some methods assume the existence of fiducial markers with known 3D geometry in the environment or object and exploit this data. Others have precomputed the 3D structure of the scene, such as the AR-View device. However, the device must be stationary, and its location must be known. If the entire scene is not known in advance, Simultaneous Localization and Mapping (SLAM) technique is used to map the relative positions of reference markers or 3D models. Structure from Motion (SfM) method is used when no assumptions can be made about the 3D geometry of the scene [26]. Another method is model-based tracking. Model-based tracking methods create a model of an object or group of objects and track the object in a real-world scene using that model. This model can be a mathematical representation that describes the object's geometry, structure, and other properties. Model-based monitoring;

- Modeling: Creating a model of the object or group of objects to be monitored.
- Matching: Making a match between the model and real-time images.
- Positioning: Determining the position of the matched model.
- Tracking: It has the feature of tracking the location and/or orientation of the object in real time.

They are techniques that explicitly use a model of the properties of tracked objects, such as a CAD (Computer-Aided Design) model or 2D template of the object, based on distinguishable features. For example, in a CAD model-based tracking

application, a mobile device's camera captures an image of a car. The app looks for matching features between the CAD model of the car and the image captured from the camera. By finding matching features, the application determines the position and orientation of the car and can integrate virtual objects into the car using this information in real time. In this way, the CAD model is used as a reference in the tracking process and helps determine the location of the real-world object.

Hand and eye tracking is technology used to detect and track the movements and positions of a user's hands and eyes in an AR environment.

- **Hand Tracking:** Hand tracking involves detecting and tracking the user's hands and fingers in real time. This can be achieved through a variety of techniques, such as depth sensors, cameras, or special hand tracking hardware. Hand tracking allows users to interact with AR content using gestures without the need for physical controllers or input devices. It provides immersive experiences where users can manipulate virtual objects with their bare hands, increasing realism and interactivity [29].
- **Eye Tracking:** Eye tracking technology tracks the movement of the user's eyes and direction of gaze. By accurately tracking the user's eye movements, AR systems can determine where the user is looking in the virtual environment. Eye tracking can be used for various purposes in AR applications, such as improving user interface design, optimizing rendering performance, or enabling gaze-based interaction techniques. For example, eye tracking can be used to adjust focus or depth of field based on the user's gaze, or to trigger actions or events depending on where the user is looking.

Combining hand and eye tracking capabilities in AR systems can further enhance the user experience by enabling more natural and intuitive forms of interaction. For example, users can manipulate virtual objects using their hands while simultaneously selecting or interacting with specific elements in the AR environment using their eye gaze [30].

2.3.2. Interaction Techniques in AR

Appropriate interaction techniques must be created for AR applications that allow end users to interact with virtual content in an intuitive way. One of the studies carried out on this subject has been concrete user interfaces (Tangible User Interfaces). It is a type of user interface that allows users to interact with digital information or virtual objects through physical, tangible objects or surfaces. Unlike traditional graphical user interfaces (GUIs), interactivity is not primarily through graphical elements on the screen but allows users to interact with digital content by manipulating and controlling physical objects.

1. **Physical Interaction:** TUIs enable users to manipulate digital content with direct physical interaction through tangible objects or surfaces. It allows users to interact with virtual content using hand gestures or body movements. Users can use gestures such as tapping, dragging, zooming, or shaking to manipulate virtual objects, navigate menus, or initiate actions in the AR environment.
2. **Body Interaction:** TUIs encourage users to interact with digital information through body movements and physical actions. This creates a more immersive and intuitive interaction experience compared to traditional input devices such as a keyboard or mouse.
3. **Mixed Reality:** Often combines physical and digital elements to create mixed reality experiences. Users can interact with physical objects or interact with digital information added to the physical environment.
4. **Physical Amenities:** Can communicate functionality or provide feedback to users using affordances of physical objects. For example, the shape, size, texture, or weight of physical objects can convey information to users about their design or functionality.
5. **Collaborative Interaction:** TUIs can support collaborative interaction, allowing multiple users to interact with digital content simultaneously using physical objects. This helps enable collaboration, communication, and shared interactions [31].

Heads-up displays (HUDs) and on-screen information provide contextual information or interactive elements placed in the user's field of view. Users can view

relevant information, navigate menus, or interact with virtual objects. HUDs and on-screen information increase situational awareness and enable seamless interaction with AR content. Voice-based interfaces allow users to control AR applications using voice commands. Users can perform actions by issuing commands, navigate menus, or interact with virtual objects. Voice-based interactions can be hands-free and especially useful in situations where users have limited mobility or need to keep their hands free.

2.3.3. AR Display Types

AR displays are various methods used to present AR content to users by allowing virtual objects to be superimposed on the real world. It aims to create an immersive and seamless integrity during the interaction of digital information with the user's physical environment.

1. **Transparent Head-up Displays (HMDs):** Also known as optical pass-through HMDs that allow seeing the real world and virtual content simultaneously. These devices use transparent or translucent displays to insert digital images into the user's physical peripheral vision. It often uses transient optics, such as waveguides, light reflectors, or mirrors, to insert virtual images into the user's line of sight. Virtual content appears to blend seamlessly with the real world, providing an interactive augmented reality experience. For example; Microsoft HoloLens, Magic Leap One, Google Glass (Enterprise Edition).
2. **Projection-based displays:** Involves projecting digital images onto physical surfaces or objects around the user. This can be accomplished using projectors or other light-emitting devices. Projected images can be static or dynamic and can interact with the physical environment or respond to user input. Users view augmented content by looking at displayed surfaces, and these surfaces serve as the canvas for virtual images. The desired virtual information is projected directly onto the physical objects to be augmented. Projection-based display is a good option for applications that don't require multiple users to wear anything and provides minimal intervention. It allows graphical information to be displayed directly on real objects and even on everyday surfaces in everyday life.

3. **Handheld Devices (Smartphones and Tablets):** Creates and presents augmented reality content using the built-in cameras, sensors and screens. AR apps running on handheld devices superimpose virtual objects onto a live camera feed of the real world and display them on the device's screen. Users interact with AR content by moving their device, moving around it, and tapping the screen; From ARKit for iOS to ARCore for Android, there are a variety of AR apps for gaming, education, shopping and more [32].

Each AR display technique offers unique benefits and suits different use cases and environments. Transparent HMDs offer hands-free interactive experiences, while projection-based AR enables large-scale interactive installations, and handheld devices offer accessible AR experiences on commonly available consumer electronics.

2.4. Deep Learning

Deep learning refers to a subfield used in the field of artificial intelligence (AI) and is one of the machine learning methods. Deep learning is designed to perform complex tasks, usually using multi-layered artificial neural networks (deep neural networks). It creates the multi-layered structure of artificial neural networks that can learn from large data sets and achieve high levels of abstraction that can be used to perform complex tasks. Deep learning uses non-linear layer like transformation and feature extraction. Each successive layer takes the output of the previous layer as input [33].

Deep learning has seen particular success in image and sound recognition, natural language processing, gaming, and other complex tasks. This success is based on large data sets and powerful computing resources. Deep learning can be effective in situations where it is difficult to manually identify features in the machine learning process, because such networks can automatically extract features from input data.

Fundamentally, deep learning is known for its ability to learn complex relationships between data through a neural network and thus usually requires large data sets and sufficient computational resources [34], [35].

With the increase in computer processing volume and the increase in the capabilities of graphics cards (graphic processors) (GPU), calculation operations have increased 1000 times in the last 10 - 15 years. With the increase in computer computational

operations, deep learning was introduced by Igor Aizenberg and his colleagues in the 2000s. Later, a large company made breakthroughs in this field [36].

Deep Learning Stages:

- Data Collection and Preparation:
 - The first step is to collect a data set suitable for a specific task. The dataset usually contains examples relevant to the task you want to learn.
 - The data set is then adapted to the learning process. This may include, for example, normalization, feature engineering or data cleaning.
- Design of the Model:
 - A deep learning model is usually a multilayer artificial neural network. This neural network consists of interconnected layers: input layer, hidden layers, and output layer.
 - The input layer receives the data that will enter the model's algorithm learning process. The output layer produces the model's predictions on a particular task.
 - Hidden layers are used to increase the learning ability of the model and extract more complex features.
- Training the Network:
 - Training the model attempts to learn to produce the expected output using the input data. In this process, weights are assigned to the network and these weights are adjusted to approximate the target output.
 - Training is usually done using an optimization algorithm. A commonly used optimization algorithm is called backpropagation.
- Verification and Testing: After the training process, validation and test datasets are used to evaluate the performance of the model. It is important to check how well the model generalizes and adapts to new data.

- **Hyperparameter Setting:** Hyperparameter tuning can be done to improve the performance of the model. This includes parameters such as learning rate, number of epochs, and network structure.
- **Deploying and Using the Model:** After successful training, the model can be deployed for use in real-world applications.

2.5. AI and AR Applications

When we look at studies combining augmented reality and artificial intelligence applications, it is clear that they are used in various fields. Trestioreanu et al [37], in radiology; In order to use the technology in live medical operating rooms, he has conducted a study based on combining a machine learning algorithm for image segmentation with the advantages of AR, especially Microsoft HoloLens. The system visualizes the segments on a Microsoft HoloLens, allowing doctors to intuitively look at holographic data instead of using 2D displays. Another area where augmented reality (AR) technology is applied is education. Lin and Chen [38] developed a deep learning recommendation system that combines AR technology and learning theories. This system is designed for students with different learning backgrounds and different majors. This study provided an image-based programming learning system with deep learning-based learning recommendations, AR technology, and learning theory to students in different learning fields to overcome undergraduate students' learning difficulties. Bermejo et al [39] have carried out studies on combining and visualizing big data with augmented reality, especially in areas such as tourism, health and public services.

2.6. Traceability of DL Models

Traceability of deep learning models generally refers to the process of understanding what's inside the model and how the model makes decisions. Traceability aims to make deep learning models more transparent, increasing their understanding and reliability. Traceability can be addressed in several ways:

1. **Input and Output Visualization:** Visualizing how the model processes inputs and produces outputs can help understand the model. Visualization of input

and output is especially important in fields such as image processing or natural language processing.

2. **Activation and Weight Visualization:** Visualizing a model's internal layers and weights can be useful for understanding how the model learns. This is used to understand what features have been learned and how they are represented.
3. **Saliency Map:** It is a technique that visualizes how deep learning models focus on a specific input and the effect of this input on the output. It is a method used especially in image classification problems. A saliency map is visualized to highlight the contribution of specific pixels of an image to the classification prediction.
4. **Gradient and Backpropagation Visualization:** Visualizing the gradients and back-propagation process during training of a model is important for understanding how the model learns. This shows how the weights are updated and how the training progresses.
5. **Decision Trees and Attribute Importance:** Techniques such as decision trees or attribute importance can explain why and how a particular prediction was made. This is especially useful for understanding the model's decisions in classification or regression problems.
6. **Sensitivity Analysis:** It is a technique used to understand the effect of the model's input characteristics on the output. This is used to understand how sensitive the model is to certain input features.

The combination of these techniques can improve the tractability of deep learning models and increase the reliability of the model. However, some deep learning models can be quite complex and difficult to fully understand. Therefore, in addition to various techniques to increase traceability, it is also important to choose simpler and interpretable versions of the model. In their study, Yu and Shi [40] presented a survey to the user regarding the visualization of deep learning models. They divided the visualization to meet the needs of different users into 4 groups, as seen in **Figure 2.6:**

- **Tools for Teaching Concepts (for beginners):** This category provides effective tools for teaching deep learning concepts to beginners and helping beginners build an intuition for how deep learning works.

- Architectural Evaluation (for practitioners): This category includes methods and tools that contribute to the visualization of network architecture.
- Tools for Debugging and Improving Models (for developers): This category includes tools to help model developers control the training process and improve models.
- Visual Description (for experts): This category includes studies focusing on the visual explanation of deep learning. This rapidly growing field of research is critical to the interpretation of deep learning.

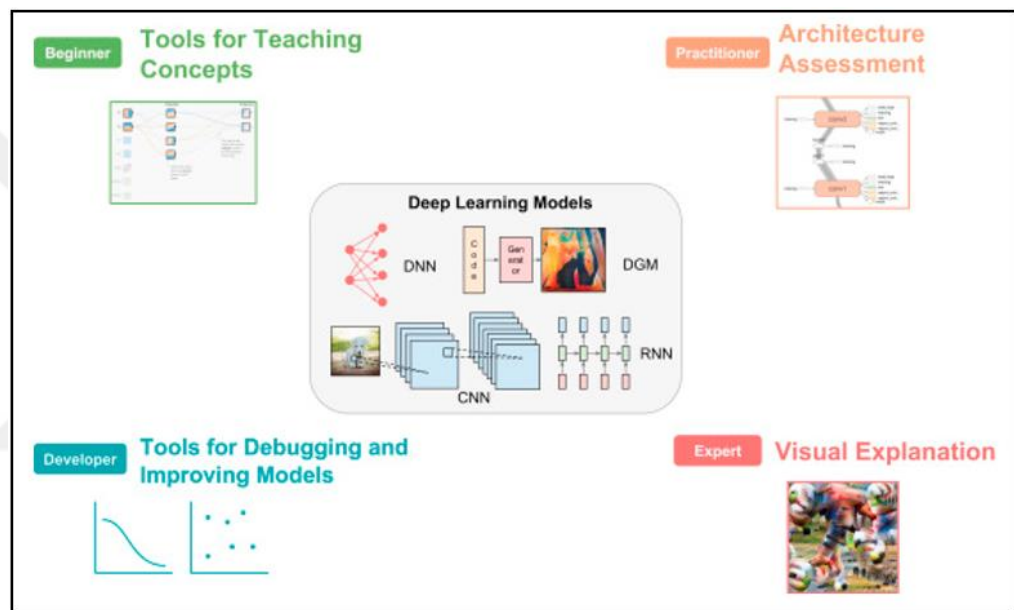


Figure 2.6: Classification of DL Visualization methods [40].

2.6.1. Visualizing Models with 2D Interfaces

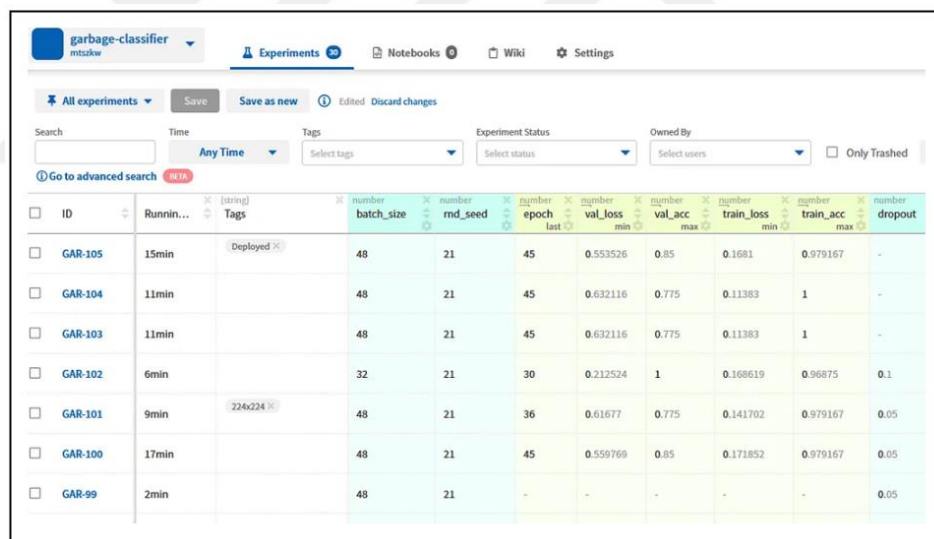
- Graphical Visualization

It is necessary to experiment with model development ideas and it becomes very important to visualize the differences between various experiments. Purposes:

- Performance tables such as ROC, Confusion Matrix,
- Overfit to predict learning curves,
- Best/worst case model predictions,

- Inferring how resource intensive model training and inference is.

Neptune is a Metadata repository for MLOps. This is used in complex machine learning projects where research and production teams run large numbers of experiments. On this platform, each project can have multiple members with different roles (viewer, contributor, administrator). This makes it easier for team members to review, share and discuss experiments. The main purpose of Neptune is to record and organize all the important data generated during the model development process. This includes a wide range of metadata, from model predictions and log losses to code changes and hardware usage. Additionally, Neptune has performance visualization features. For example, metric graphs such as a ROC curve or confusion matrix can be recorded during or after training. These charts provide important information to evaluate the model's performance and make decisions (**Figure 2.7**).



ID	Runnin...	Tags	number batch_size	number rnd_seed	number epoch last	number val_loss min	number val_acc max	number train_loss min	number train_acc max	number dropout
GAR-105	15min	Deployed	48	21	45	0.553526	0.85	0.1681	0.979167	-
GAR-104	11min		48	21	45	0.632116	0.775	0.11383	1	-
GAR-103	11min		48	21	45	0.632116	0.775	0.11383	1	-
GAR-102	6min		32	21	30	0.212524	1	0.168619	0.96875	0.1
GAR-101	9min	224x224	48	21	36	0.61677	0.775	0.141702	0.979167	0.05
GAR-100	17min		48	21	45	0.559769	0.85	0.171852	0.979167	0.05
GAR-99	2min		48	21	-	-	-	-	-	0.05

Figure 2.7: Neptune is an open source library that allows users to regularly save the data they create during the model development process [41].

WandB (Weights & Biases) is a platform used in the development process of deep learning models, and the visualization techniques. W&B's visualization techniques include:

- Training Metrics Graph: Graphs showing the change of training metrics over time to monitor the performance of the model during the training process. These

graphs are used to understand how metrics such as accuracy, loss, precision, recall change throughout training.

- **Distribution of Model Parameters:** Graphs showing the distribution of the learned parameters of the model. This allows visual inspection of how weights or biases change and distribute during training.
- **Visualization of Model Predictions:** Visualization of the predictions made by the model on real data. For example, in classification problems, a confusion matrix or ROC curve is visualized to compare predicted classes with actual classes.
- **Visualization of Hyperparameter Tuning:** Graphs or tables comparing the performance of different hyperparameter tunings or model structures. This is used to determine the best model structures or hyperparameter settings.
- **Visualizing the Architecture of the Model:** Graphs or diagrams that show the architecture of the model (for example, connections between layers). This is used to better understand how the model is structured (**Figure 2.8**).

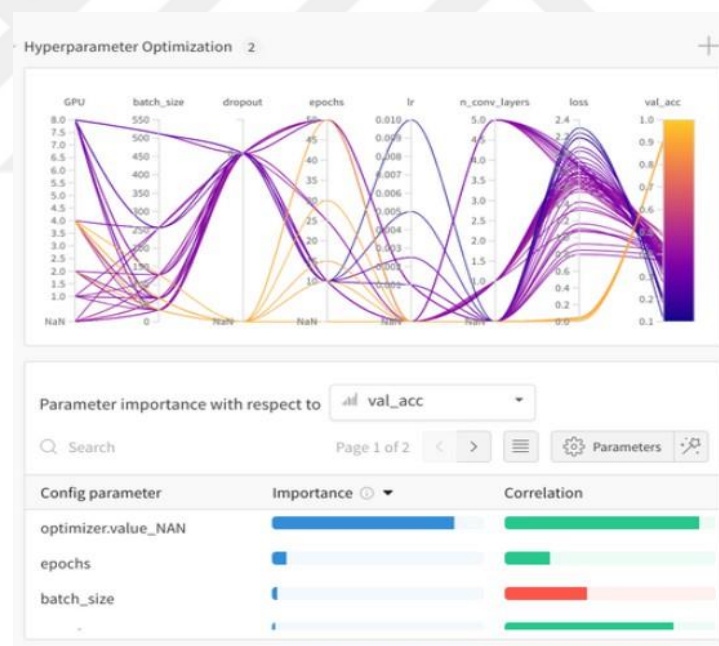


Figure 2.8: Wandb is very effective in monitoring and understanding the performance of the model [42].

TensorBoard is TensorFlow's visualization tool. Metrics and graphs created during model training are used to visually examine and analyze data such as training and validation losses (**Figure 2.9**).

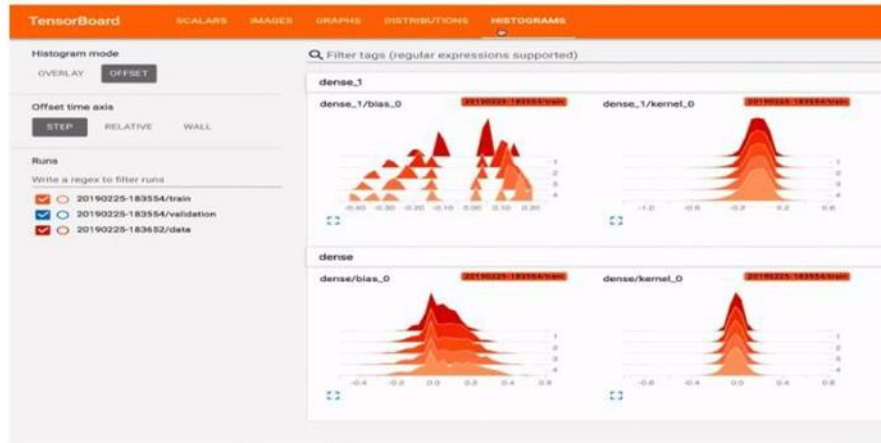


Figure 2.9: Tensorboard is used to monitor the model’s performance, detect errors, and help improve the model [43].

Visdom is a visualization library developed by Facebook and is used specifically for training deep learning models and visualizing their results. Features that Visdom provides for deep learning visualization include various graphs, tables, heatmaps, scatter plots, tracking training metrics, and visual comparison of model results. Thanks to these features, it becomes possible to better understand the training process of deep learning models and monitor model performance (**Figure 2.10**).

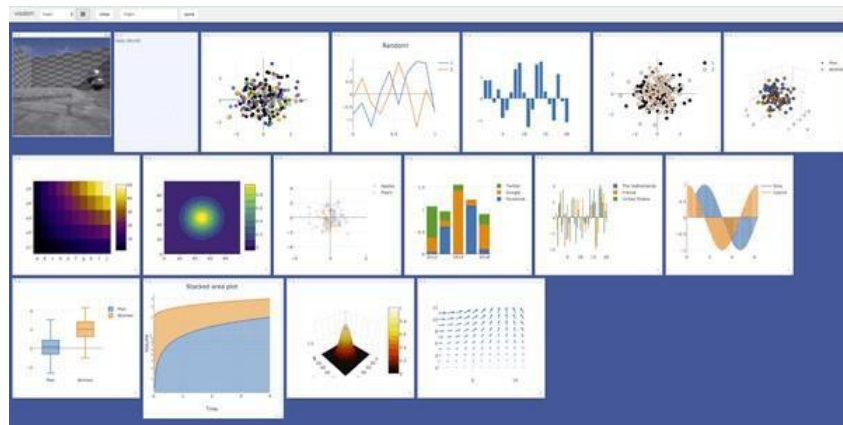


Figure 2.10: Visdom is a library that enables interactive visualization of data through a browser-based interface [44].

HiPlot is an open-source tool used to visualize machine learning experiments (**Figure 2.11**). The main features of HiPlot are:

- **Interactive Discovery Tools:** HiPlot provides interactive discovery tools to discover and understand relationships between experiments. Users can easily compare different hyperparameter settings and results between experiments.
- **Visualization of Parameter Spaces:** HiPlot is used to visualize the distributions of hyperparameters used in experiments. This is important for understanding the effects of hyperparameters and determining the best settings.
- **Visualization of Results:** Used to visualize the results of experiments (e.g. accuracy, loss). This is important to compare performance between experiments and obtain the best results.
- **Labeling and Filtering:** HiPlot allows you to label and filter between experiments. This makes it easier to explore by focusing on specific experiments.
- **Export and Share:** HiPlot enables export and sharing of visualizations. This makes it easier to collaborate and share results with others.

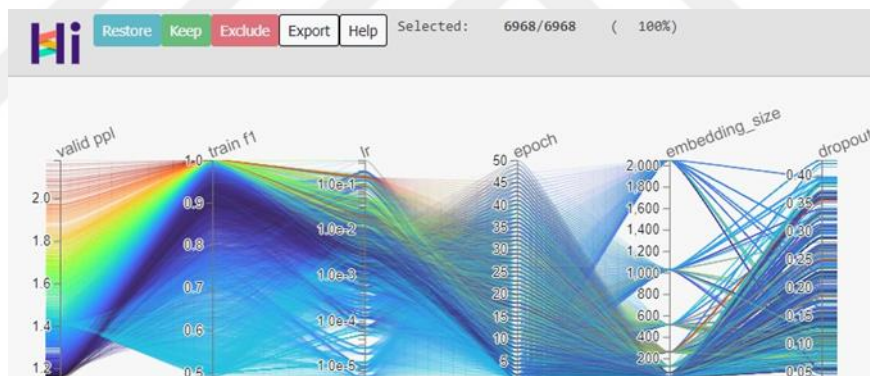


Figure 2.11: Hiplot is generally used for hyperparameter optimization and results exploration [45].

- **Visualization of Network Structure**

Purposes:

- Mastering the layers by analyzing the network,
- To provide an easy indication of tuning hyperparameters to provide greater accuracy and lower error rates,
- Providing accurate data for further action on it,

- Helping realize and recognize the dataset, which can help identify corrupt data, outliers, and more.

CNNVis is a new visual analytics system that will help machine learning experts better understand, diagnose and improve CNNs (**Figure 2.12**).

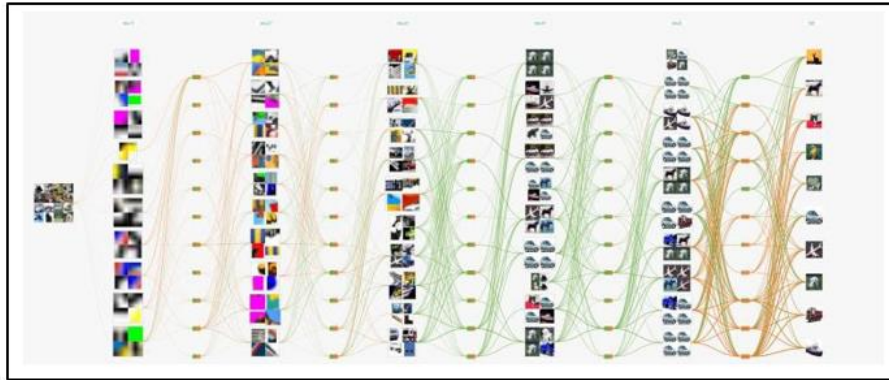


Figure 2.12: CNNVis is a tool that focuses on analyzing a snapshot of the CNN models [46].

Neural Networks Playground makes neural networks a little more accessible and easier to learn (**Figure 2.13**).

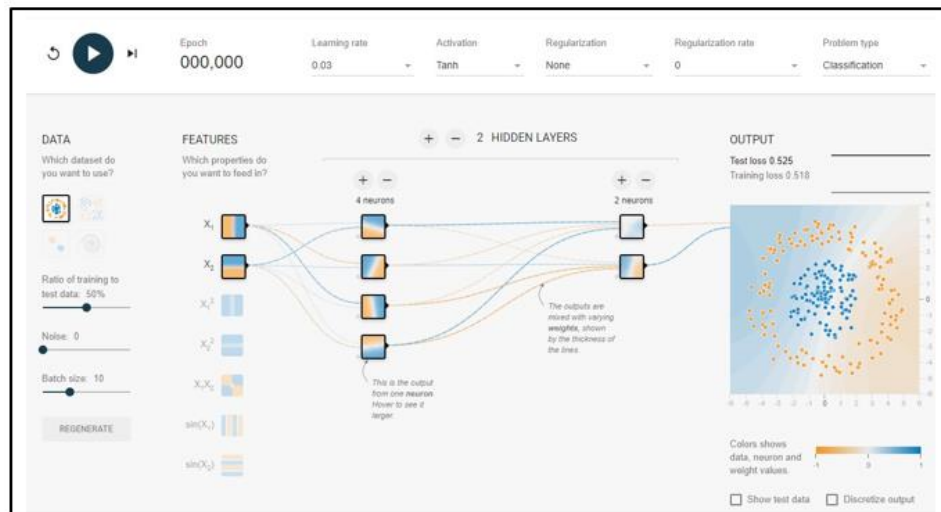


Figure 2.13: Neural Networks Playground is a simple tool for NN model exploration and visualization [47].

Netron is a viewer for neural network, deep learning and machine learning models (**Figure 2.14**).

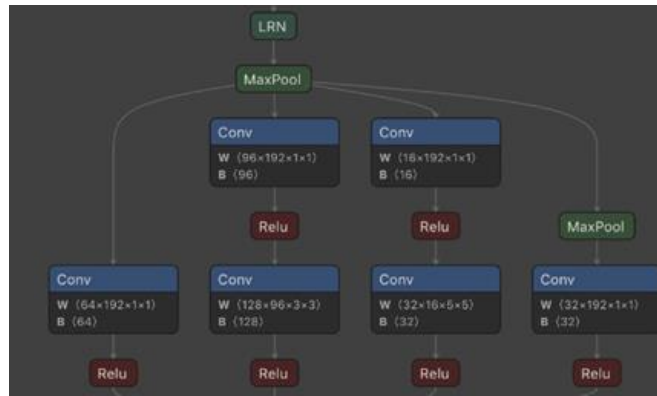


Figure 2.14: Netron provides a simple and intuitive interface for visualizing models in various formats [48].

ANN Visualizer provides visualization by creating a presentable graph of the neural network being created (**Figure 2.15**).

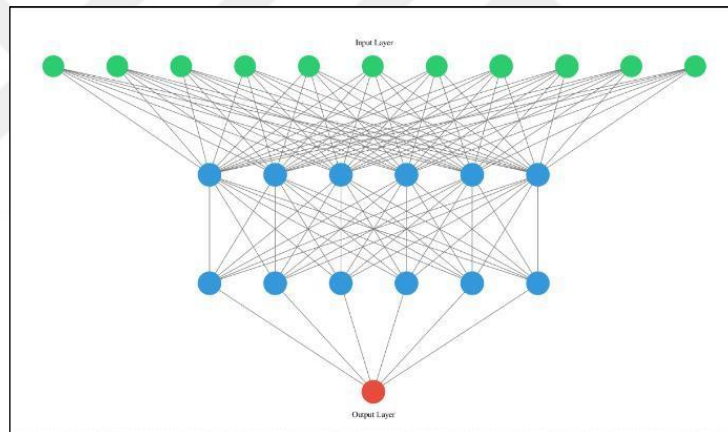


Figure 2.15: ANN Visualizer is a python library for visualization NN [49].

DeepGraph, is a tool offered by PyCharm for understanding and visualizing deep learning models. This tool helps to understand the complex structure of deep learning models more easily by using a combination of visualization and code matching features. DeepGraph is a powerful tool for understanding and developing deep learning models and works in interaction with Python code thanks to the rich integration of PyCharm. In this way, it helps to be more efficient in the model development process [30].

- Feature Visualization

There is a growing perception that people need to interpret the results of deep learning and eventually open the black box. It is a technique used to understand the basic features learned by a deep learning model and the relationship of these features to input data. It is especially widely used in the field of image processing and plays an important role, especially in deep learning models such as convolutional neural networks (CNNs). Examining the feature maps of each layer of the model is important to understand how the features the model has learned are represented. Feature maps show the activations of each layer and how it interacts with input data. Deep architectures have demonstrated state-of-the-art results in various environments, especially image datasets. Beyond model descriptions and quantitative analyses, there is a need for qualitative comparison of solutions learned by various deep architectures. In their study, Erhan et al [50] aimed to find good qualitative interpretations of the high-level features represented by such models. *Activation Maximization* aims to create an image that maximally activates a particular neuron, thereby revealing the features the neuron has learned during training. This method starts from a random image and updates the image during iterations of gradient ascension to increase the activation of a particular neuron. Activation maximization is a technique used to generate images that maximally activate specific neurons or filters within a neural network. The goal is to understand the patterns or features to which these neurons are sensitive by visualizing the input stimuli that lead to maximal activation of a particular neuron or its filter. On the left in **Figure 2.16**; Visualization of 144 units from the first (column 1), second (column 2) and third (column 3) hidden layers of a DBN (top) and an SDAE (bottom) using the hidden layer activation maximization technique is shown. On the right: 4 examples of solutions to the optimization problem, subject to 9 random generation, for units in the 3rd layer of SDAE.

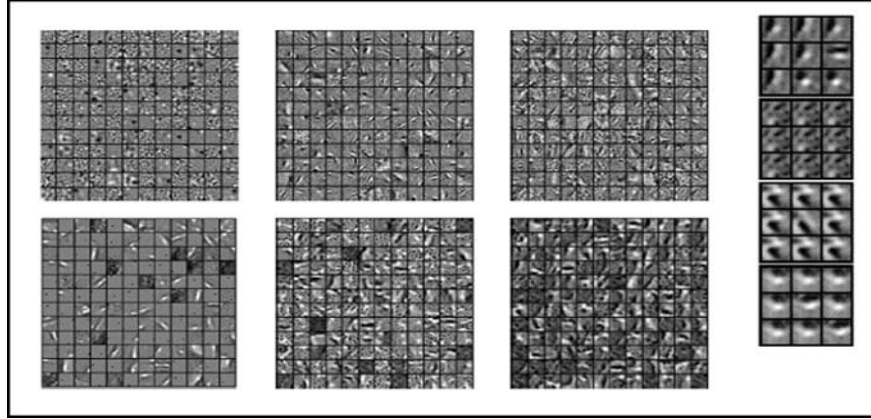


Figure 2.16: Visualization with activation maximization method [50].

Another important study is the saliency map, also called sensitivity map. Simonyan et al. [51] discussed a visualization technique based on calculating the slope of the class score according to the input image. It calculates a class saliency map specific to a given image and class. They have shown that such maps can be used for weakly supervised object segmentation using ConvNet classification. Finally, they established the connection between gradient based ConvNet visualization methods and deconvolutional networks.

To analyze the [52] network, Springenberg et al introduced a new variant of the “deconvolution approach” to visualize features learned by CNNs, creating a framework that can be applied to a broader network structure than existing approaches. This technique can also be applied to the saliency map. This method adds an additional signal when calculating the single backpropagation pass to obtain a sharper and clearer saliency map. However, there is still quite a lot of noise in the guided backpropagation saliency map. To increase salience, Smilkov et al. [53] add pixel-wise Gaussian noise to many copies of the image and average the resulting gradients. Experiments show that this technique can significantly denoise the saliency map. The comparison between these different methods is shown in **Figure 2.17**.

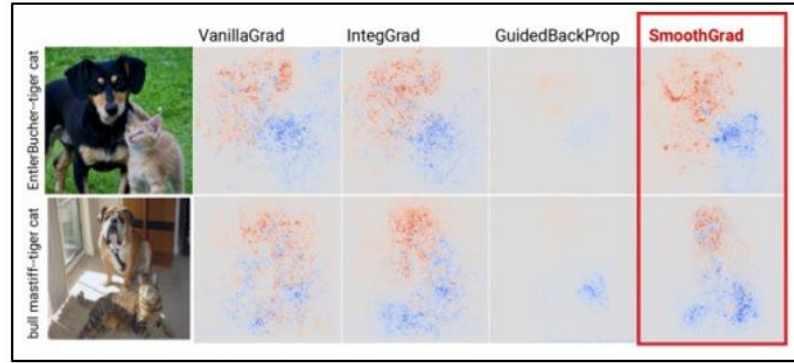


Figure 2.17: Discriminateness of different methods for calculating the saliency map [53].

2.6.2. Visualizing Models with 3D Interfaces

Looking at the visualization of artificial intelligence models; Inkarbekov et al. [54] discusses the investigation of virtual reality as a tool for visualizing artificial intelligence systems in the context of human computer interaction. It also highlights various methods in the field of AI visualization, highlighting the power of interactive 3D visualization and other specialized approaches in making information about AI models more accessible and engaging.

Bock and Schreiber made model visualization in virtual reality using the unreal engine (**Figure 2.18**) [55].

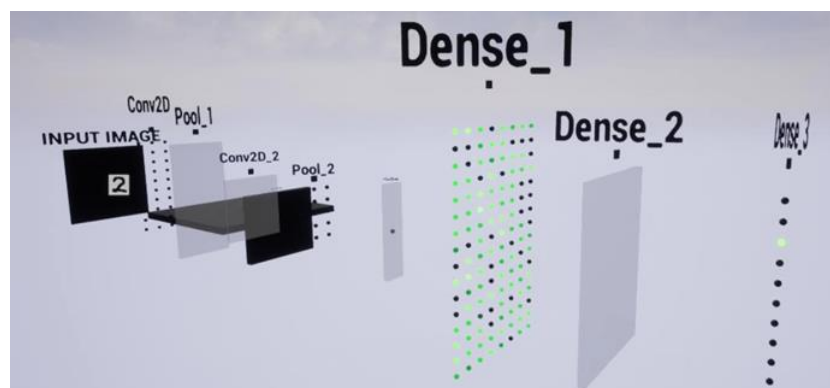


Figure 2.18: Network visualized with VR in Unreal Engine [55].

Yosinski et al. [56] created the Deep Visualization Toolbox, shown in **Figure 2.19**, based on the development of better tools for visualizing and interpreting neural networks to understand how models work and what calculations they perform in the

intermediate layers. This open-source software allows users to examine CNNs with an image or a live webcam to see how each neuron responds to input. The software consists of several parts that display the input image, activations, deconvolution visualizations, and images from the training set that maximally illuminate the input image.

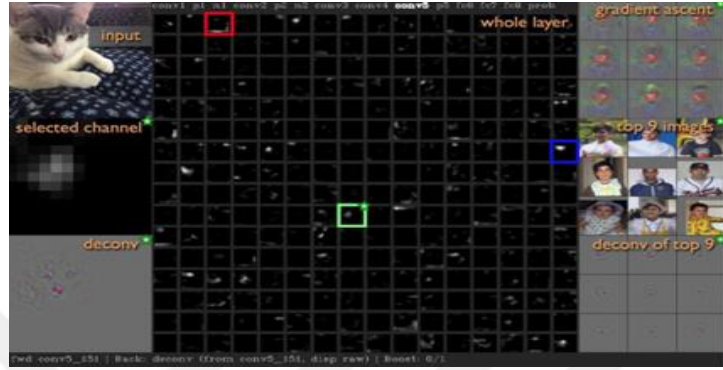


Figure 2.19: Deep Visualization Toolbox is used to create neuron-by-neuron visualizations using regular optimization [56].

Kath and Lüers et al. [57] created a virtual reality tool to automate the process of assigning data entries to different categories. In addition to moving around the virtual space, the user can also use hand gestures to assign parts of the cloud to different groups reflecting their class labels (i.e. digit ID), in other words, to annotate the data. The topological organization and movement patterns observed in virtual space are direct and tangible results of the functioning of the underlying deep neural network (**Figure 2.20**).

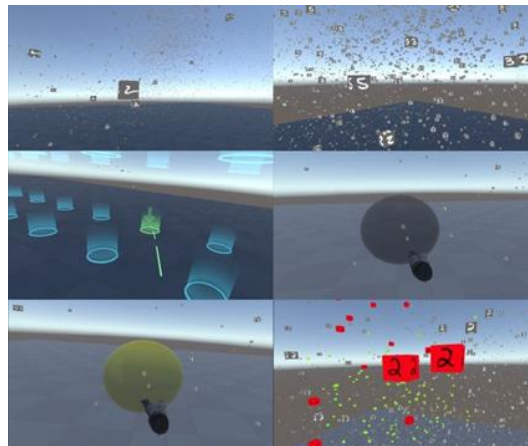


Figure 2.20: Steps of the description process from the user's perspective in virtual space and model architecture [57].

Linse, Alshazly, and Martinetz [58] applied visualization in virtual reality by addressing the problem of how to create complex, large-scale CNNs. Thus, they enabled popular architectures to be visualized in a 3D computational graph with tens of thousands of feature maps and branches. They have also introduced a new Pytorch module to dynamically connect Pytorch to Unity, providing developers and researchers with a convenient interface to visualize their architectures (**Figure 2.21**).

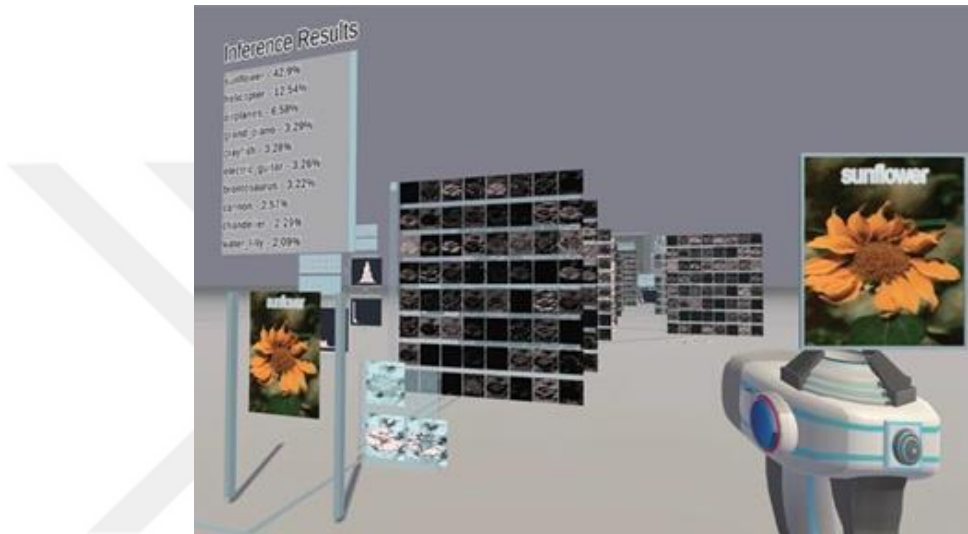


Figure 2.21: DeepVisionVR application allowed the user to freely navigate the layered network, retrieve and place images, move and scale layers for better readability, perform feature visualization, and export the results [58].

Aamir et al [59] have presented a new approach to visualize and interpret deep networks in VR in an immersive way, where the user can move freely within an AlexNet (**Figure 2.22**).

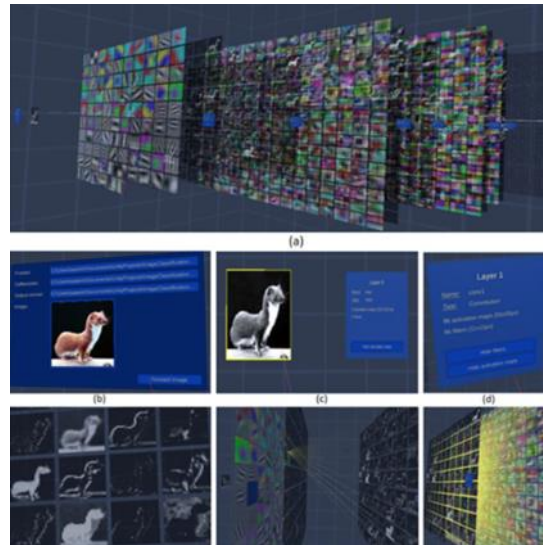


Figure 2.22: This visualization technique offers exciting possibilities for understanding and interacting with complex models in immersive environments [59].

Van Horn et al. [60] have created a comprehensive deep learning environment, as shown in **Figure 2.23**, that allows the user to train and test networks of up to 10 layers in VR. The tool is not intended as a general analysis framework for arbitrary CNN architectures, rather, it is designed as a bridge that allows people with less knowledge of computational science to perform deep learning.

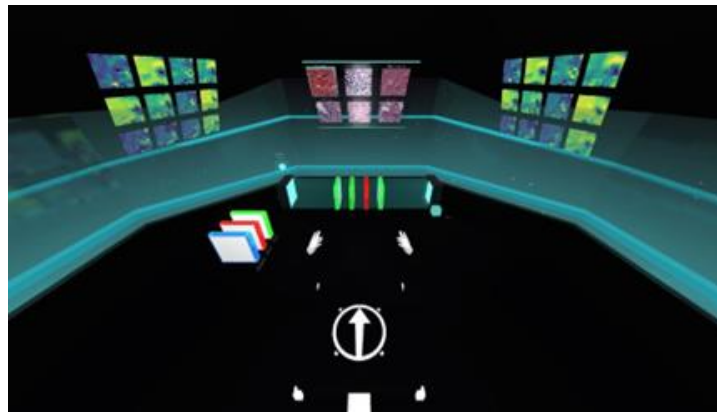


Figure 2.23: It is a VR visualization tool specifically targeted at individuals with limited computational science knowledge to perform deep learning tasks [60].

3. METHODOLOGY

In this section, the application part of the study will be explained. The stages of the study is shown in **Figure 3.1**.

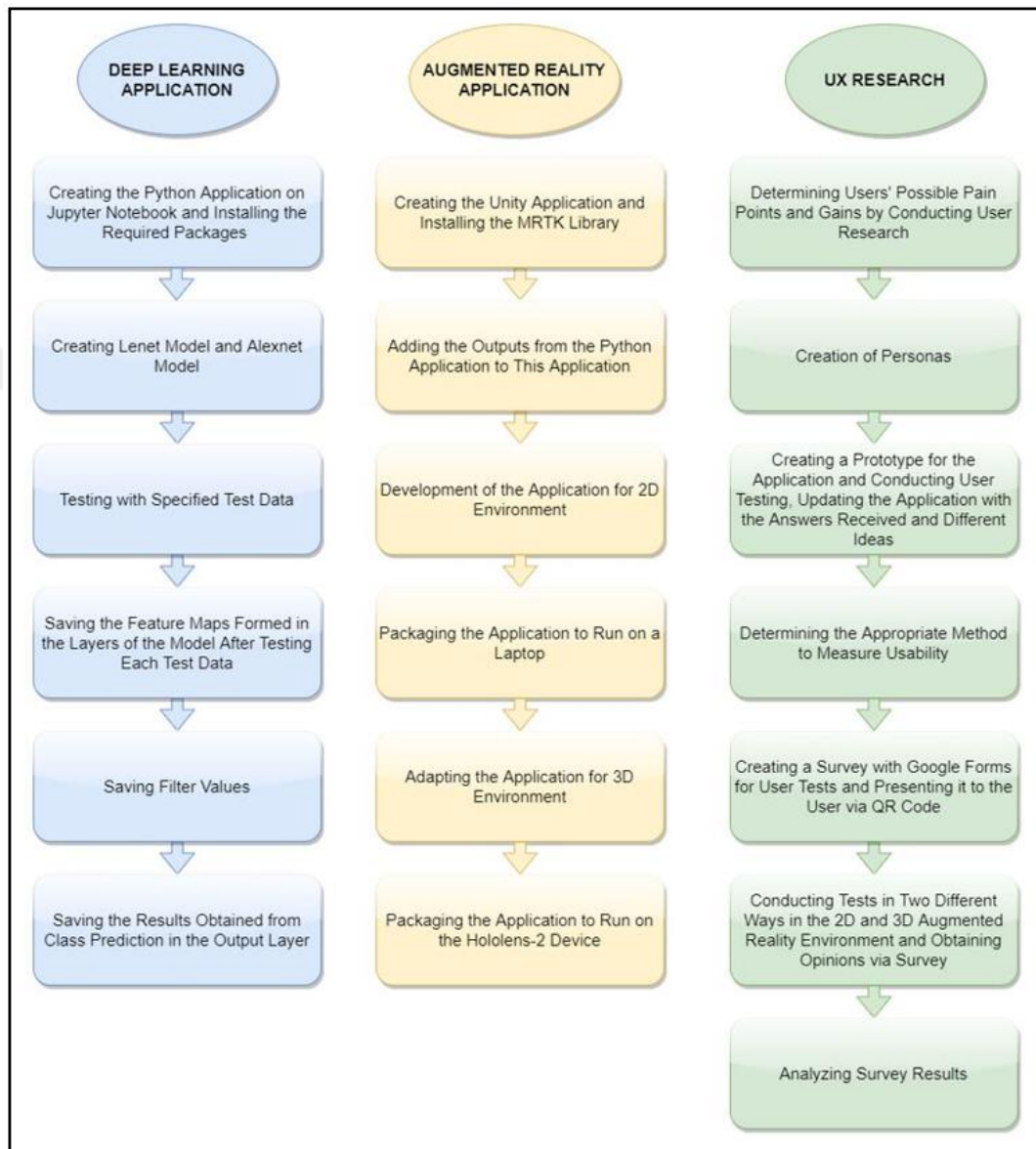


Figure 3.1: Parts and flow diagram of the study.

3.1. DL Model Creation

Two ready-made models, Lenet and Alexnet, were used in the study. Models were created with Python language on Jupyter Notebook.

- Lenet Model

Yann LeCun [61] and others, one of the pioneers of deep learning, found LeNet-5 in their work in 1998. Based on the MNIST dataset, banks have used LeNet-5 to recognize handwritten checks. It reviews various methods applied to handwritten character recognition and compares them to a standard handwritten digit recognition task. Convolutional neural networks specifically designed to deal with the variability of 2D shapes have been shown to outperform all other techniques. The architecture of the Lenet model is given in **Figure 3.2**.

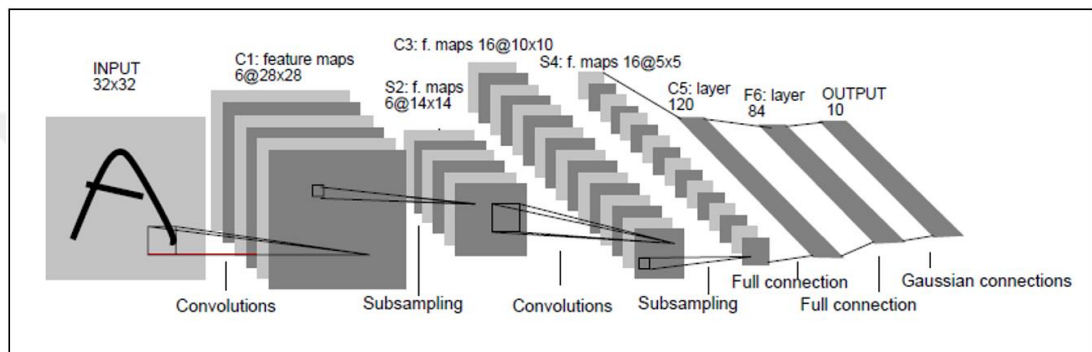


Figure 3.2: Lenet architecture [61].

In the CNN architecture, it is passed through the first convolutional layer. Each subsequent convolutional layer aims to discover relationships between pixels and neighboring pixels by using kernels to extract various features from the image. CNN typically consists of a convolutional layer that applies a series of learnable filters to the input image. These filters detect various features such as edges, textures, and patterns. The output of the convolution layer is a feature map that highlights the presence of these features in the image. A common choice after the convolutional layer is to use a pooling layer such as maximum pooling or average pooling. This layer reduces the spatial dimensions of the feature map while preserving the most important features. Pooling helps subsample data, reducing computational complexity and extracting the most relevant information.

After the pooling layer, additional convolution and pooling layers can be stacked to capture increasingly complex features in the image. This hierarchical structure allows the network to learn abstract representations of input data. To add nonlinearity to the model and increase its expressiveness, activation functions such as ReLU (Rectified

Linear Unit) are applied after each convolution layer. ReLU sets negative values to zero, preserves positive values, and offers nonlinear transformations. Towards the end of the network, fully connected layers are often used. These layers process the extracted features and make predictions based on them. Fully connected layers connect all neurons in the previous layer to neurons in the current layer, allowing the network to learn complex relationships between features and output labels. Finally, the output layer of the CNN depends on the specific task at hand. For example, in a classification task, a SoftMax layer can be used to generate probabilities for each class, while in a segmentation task, a convolutional layer with a pixel-wise activation function can be used. It is important that the architecture and order of layers in a CNN can vary depending on the specific requirements of the task and the dataset used. Moreover, researchers often experiment with different architectures, hyperparameters, and regularization techniques to optimize performance and obtain accurate and reliable results [62].

LeNet is typically used for simple visual classification problems such as digit recognition. Basically, it has an architecture that includes convolution and sampling layers. These layers are used to extract features from the input images and reduce their size. The basic architecture of LeNet is as follows:

1. Convolutional Layer: Feature maps are created by shifting filters on the input image.
2. Pooling Layer: It is used to reduce the size of feature maps and reduce the computational load.
3. Fully Connected Layers: It is used to classify from final feature maps.
4. Activation functions: Non-linear features are added to the model with incoming data.

In the application, the model was created using the Tensorflow keras library. Mnist data set was used. The architecture of the model is as seen in **Figure 3.3**.

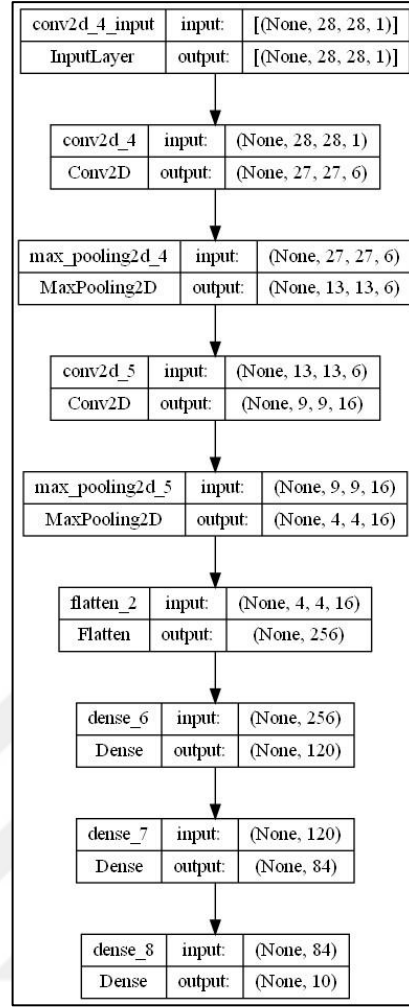


Figure 3.3: Architecture of the lenet model used in this study.

```

conv2d [[[[[ 0.1436145  0.36392194 -0.35958698  0.2831421  0.42979705
-0.17651269]]

[[ 0.41274172  0.5671656  0.31445003  0.40183255  0.23169805
-0.06156456]]]

[[[ 0.37006384  0.33062437  0.3566598 -0.17446826 -0.11316572
0.24071108]]

[[[-0.3338682  0.2797875  0.4647417 -0.1130608  0.47212687
0.58496  ]]]]

conv2d_1 [[[[[-0.0035638  0.01857955  0.00345579 ... -0.08340353  0.03945175
-0.05052324]
[-0.11797303 -0.02329039  0.06379768 ... -0.02704247 -0.03722082
0.01436257]
[-0.04601551  0.00257322 -0.04086462 ... 0.11793681 -0.03177359
0.01895607]
[-0.10492108  0.04813968 -0.01049864 ... -0.01369665  0.01005093
-0.02693376]
[ 0.00143564 -0.00890596 -0.04392277 ... 0.02679621  0.1227518
-0.01257837]
[ 0.03702086 -0.10714524 -0.09415971 ... -0.018928 -0.03784878
0.03388617]]

.
.
.

```

Figure 3.4: Values of the filters in convolution layers recorded for visualization in this study.

- Alexnet Model

AlexNet is a convolutional neural network architecture that gained significant attention after winning the ImageNet Large Scale Visual Recognition Challenge, a large-scale image recognition competition, in 2012. Developed by Alex Krizhevsky, Ilya Sutskever and Geoffrey Hinton [63], it broke new ground in the field of deep learning and demonstrated the potential of deep neural networks for image classification tasks. The network contains 8 layers (the first 5 are convolutional and the remaining 3 are fully connected). The output of the last fully connected layer is fed to the 1000-way softmax, which produces a distribution over 1000 class labels. Response normalization layers follow the first and second convolution layers, while pooling layers follow both the response normalization layers and the fifth convolution layer. It applies dropouts with probability 0.5 on the last two fully connected layers in the network. This technique reduces the complex co-adaptations of neurons because a neuron cannot rely on the presence of certain other neurons. It is therefore forced to learn more robust features that are useful along with many different random subsets of other neurons [63].

In the study, the model is created as pre-trained from the Pytorch library. This model is a pre-trained model using weights pre-trained with the ImageNet dataset. This training allows the model to learn the feature vectors used to represent image data. Training a model like AlexNet on ImageNet is important to improve general object recognition capabilities, learning visual features. The input size of this model trained with the ImageNet dataset is 224x224 and the number of classes is 1000. The architecture is specified in **Figures 3.7**.

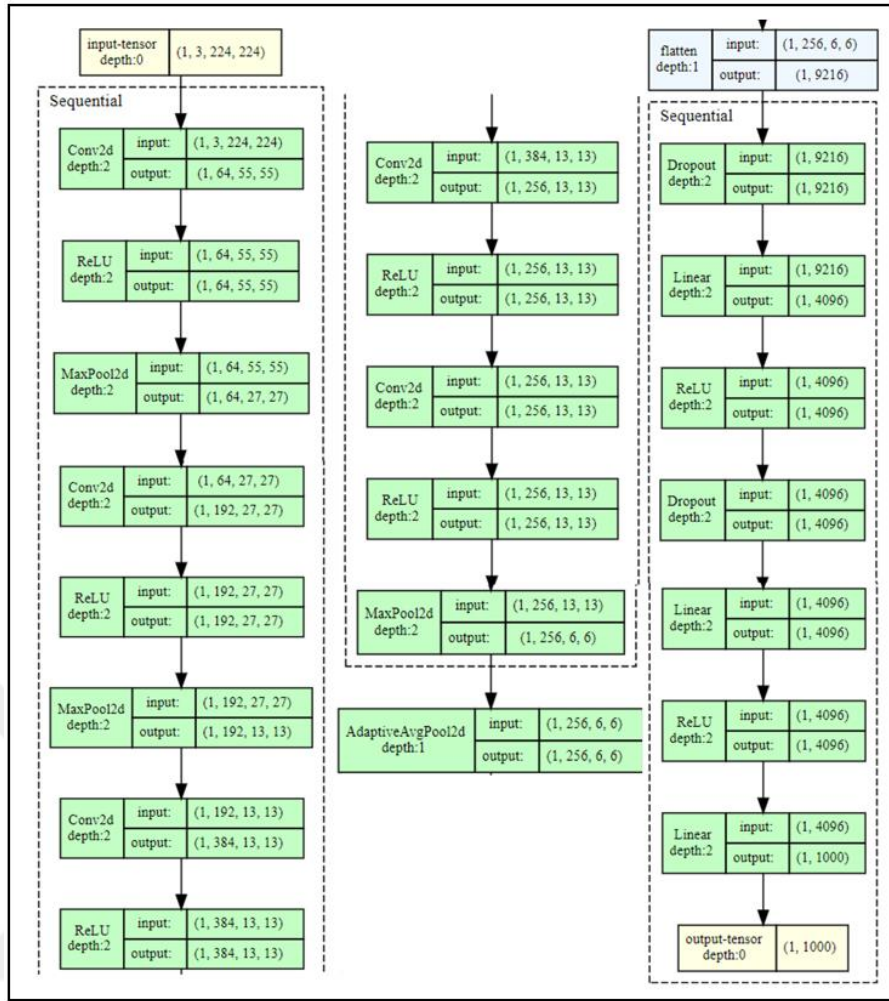


Figure 3.7: Architecture of the alexnet model used in this study.

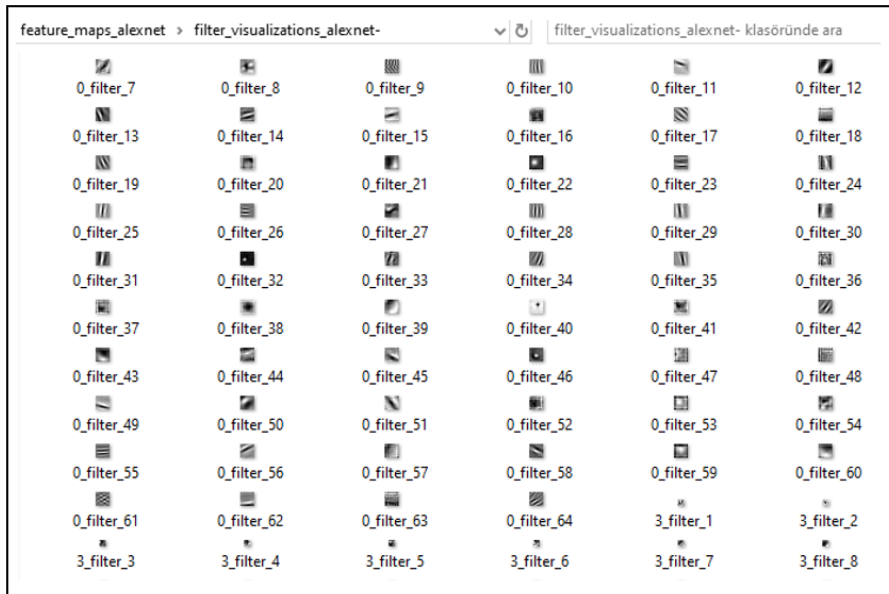


Figure 3.8: Filters of the alexnet model used in the study are saved as image files.

In the trained model, data are given to the model one by one for prediction. The information of the filters used at this stage was recorded as seen in **Figure 3.8**. The feature map of each image was visualized for all layers and recorded as seen in **Figure 3.9**. In addition, all prediction results, especially the 5 highest results, were recorded.

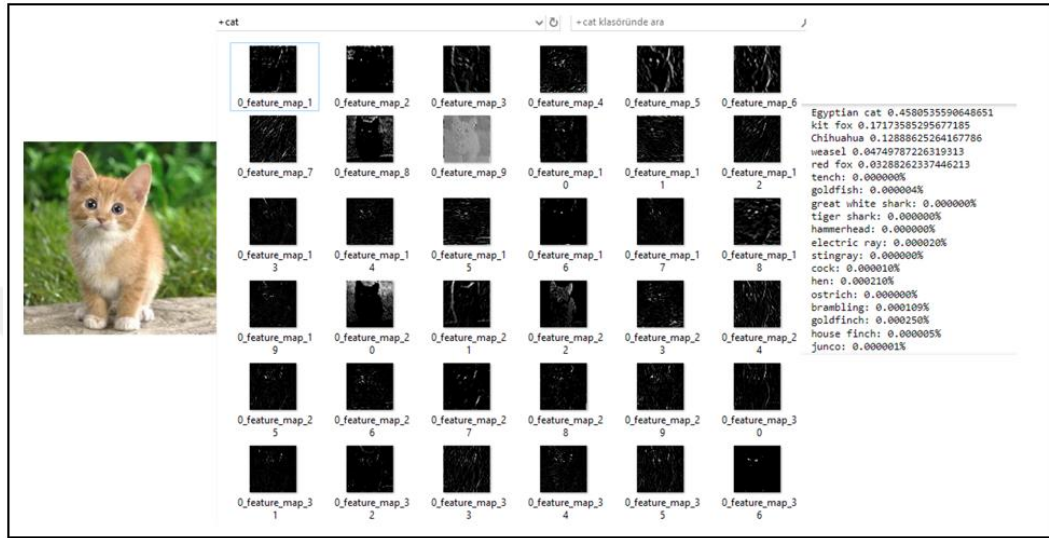


Figure 3.9: Giving the cat image to the model, extracting the feature maps and the result.

3.2. DL Models Visualization Application

Unity is a popular game engine used to develop games and interactive 3D experiences. Not only games, but also educational simulations, virtual reality experiences, augmented reality applications and many more can be developed with Unity. In this study, the visualization of the models was done with the Unity game engine. It was preferred because it provides library usage support for the Hololens-2 device to be used.

3.2.1. Visualization in 2D Environment

- Unity application is created.
- A 2D scene is created.
- Information and images of all feature maps and filters taken from the Python application are added to this application for use.

- A menu is added for Lenet or Alexnet Model selection.
- When the application is opened, the image objects of the feature maps to be placed on the screen are created under the relevant layers. However, it is not active. A structure as in **Table 3.1** is established and when the layer is opened from the model selected from the menu, the relevant images will be active, as seen in **Figure 3.14**, **Figure 3.15** and **Figure 3.16**.

Table 3.1: The content of the object created for layers in the Unity scene.

Lenet Layers	Alexnet Layers
Conv ₁	Conv ₁
MaxPool ₁	MaxPool ₁
Conv ₂	Conv ₂
MaxPool ₂	MaxPool ₂
flatten	Conv ₃
dense ₁	Conv ₄
dense ₂	Conv ₅
output	MaxPool ₃

In this structure, all layers are objects with empty image objects inside them, as in **Figure 3.10**. When the application starts, the empty image objects inside these main objects are filled one by one with the images taken from the relevant folder.



Figure 3.10: Created image objects for feature maps on layers for the Lenet Model.

- Filters are created in the same way.

- In this case, the layers of the application will look like **Figure 3.12** before opening, but after opening they will look like **Figure 3.14**.
- The application has been packaged and made ready for use so that it can run on the computer.
- The stages of the application will be as follows: The user selects the model type from the Menu (**Figure 3.11**). The selected model opens.

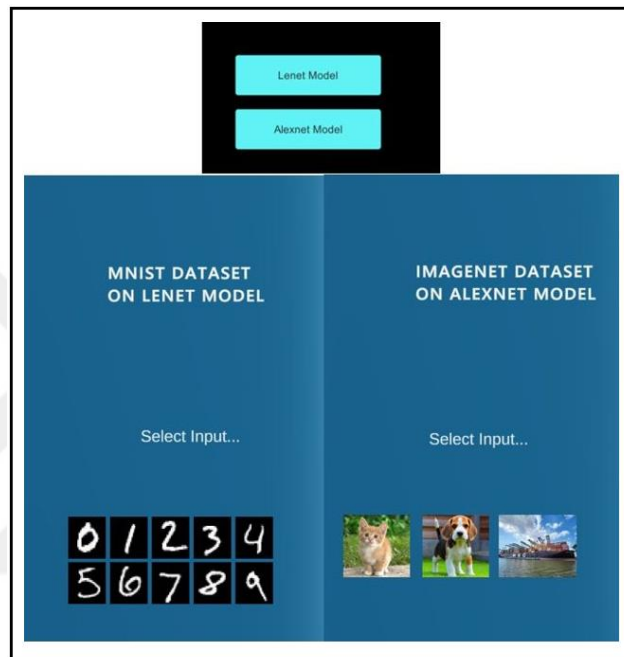


Figure 3.11: Menu and screens that open when selection is made.

The user selects an input from the test data. The feature maps of the selected image appear in layers (**Figure 3.12**, **Figure 3.13**).

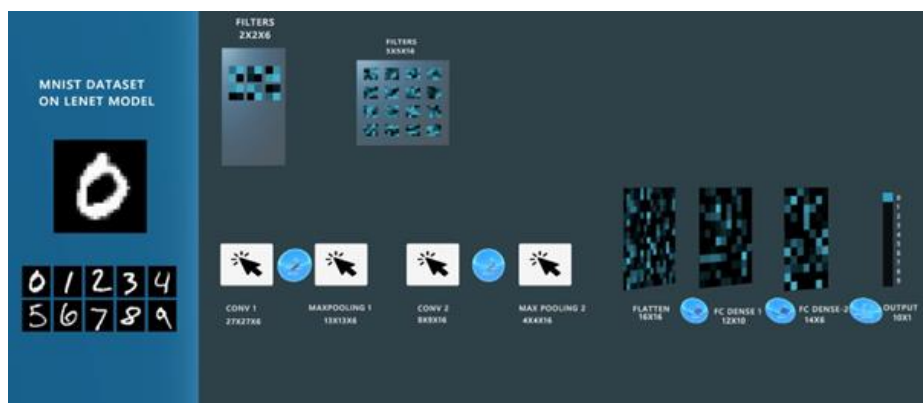


Figure 3.12: Layers opened after selecting image "0".



Figure 3.13: Layers opened after selecting the cat image.

The user opens and examines the feature maps by clicking on individual layers (Figure 3.14, Figure 3.15).

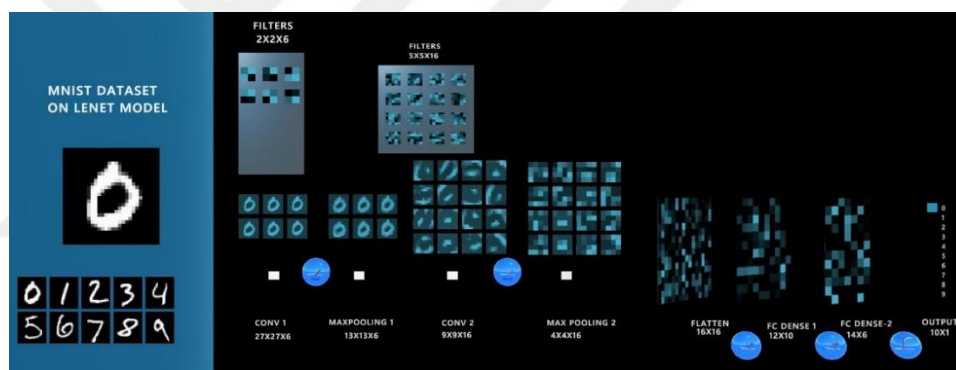


Figure 3.14: The content of the layers after clicking the buttons.

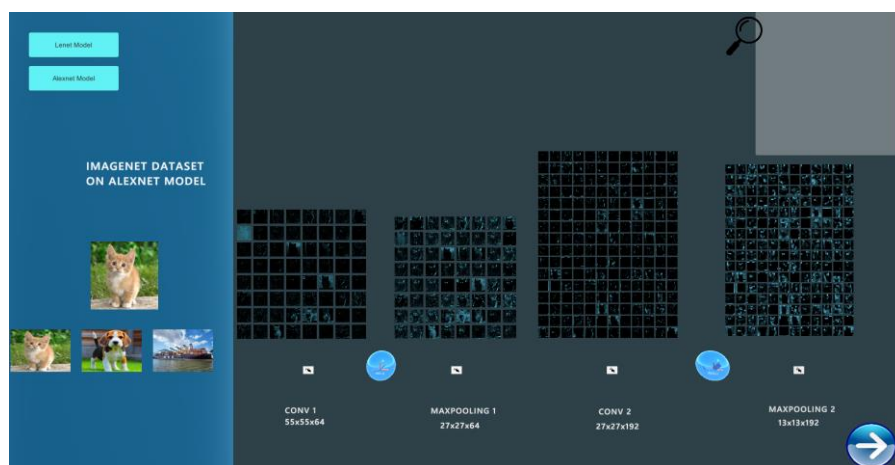


Figure 3.15: The content of the layers after clicking the buttons.

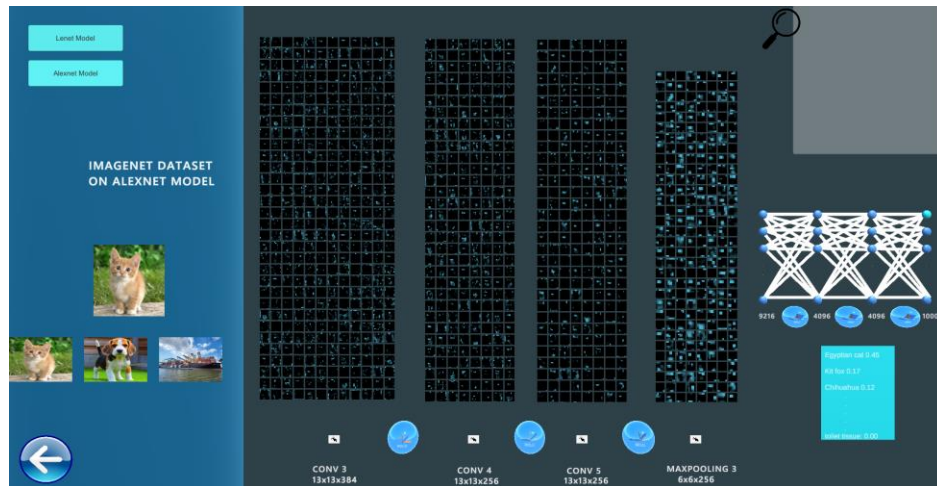


Figure 3.16: Viewing feature maps selected from layers.

- When selecting another entry, the images within the layers are closed and the user must click on the layer buttons again.
- It examines the prediction phase of this different input it selects with the updated feature maps.

3.2.2. Visualization with AR Interfaces in 3D Environment

Visualization in a 3D environment will be provided with Microsoft HoloLens-2 glasses using augmented reality technology. Looking at the features of the HoloLens-2 device; Hand tracking; Using the AHAT depth camera, HoloLens 2 can capture hand movements to obtain hand tracking data. This camera is designed to detect the user's hands in three dimensions and forms the basis of HoloLens 2's hand tracking feature. It is specifically designed to precisely track the user's hands in terms of depth, position and movement. This camera constantly detects the position of the fingers and hand and processes this information through special algorithms on the HoloLens 2's processor. It allows the user to interact with virtual objects in a more natural way. Users can hold, turn, push and pull holographic objects by moving their hands. In this way, the augmented reality experience becomes more impressive and realistic.

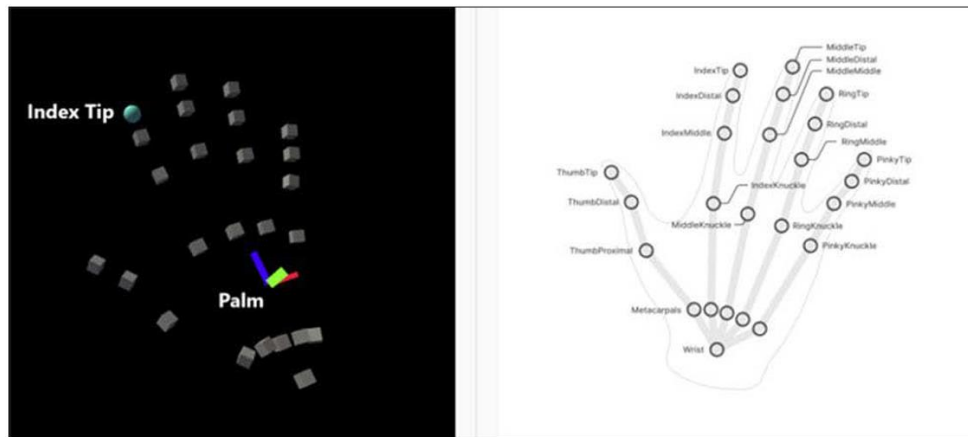


Figure 3.17: The hand tracking system uses people's palms and fingers as input [64].

Eye tracking; IR (Infrared) Cameras or Sensors detect the user's eyes and monitor their eye movements. These cameras often use infrared light to detect the position, direction of gaze, and movement of the eyes. Spatial mapping in HoloLens refers to the process by which the device detects the physical world around it and creates a digital map. This process allows users to interact with virtual objects with the real world and enrich the augmented reality experience. This mapping is provided by various sensors such as depth cameras, RGB cameras and IMU (Accelerometers and Gyroscopes) on the front of the HoloLens. These sensors are used to detect the position of the device and the location of surrounding objects [64]. Head tracking (**Figure 3.17**): 4 visible light cameras. Eye tracking: 2 Infrared (IR) cameras. Depth: 1 MP Time-of-Flight depth sensor. Inertial measurement unit (IMU): Accelerometer, gyroscope, magnetometer. Camera: 8 MP stills, 1080p30 video (**Figure 3.18**).

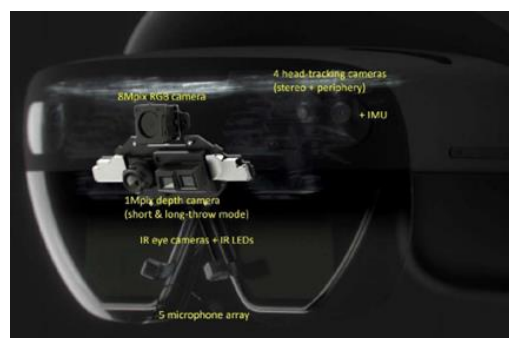


Figure 3.18: HoloLens-2 sensors [64].

Stages of Application:

- A 3D scene is created.
- MRTK library is added to use the features offered by the Hololens device. Mixed Reality Toolkit (MRTK) is an open-source library developed by Microsoft and is used to develop augmented reality (AR) and virtual reality (VR) applications. MRTK can integrate with popular game engines such as Unity and Unreal Engine. MRTK's main goal is to provide developers with a set of ready-made components and tools that they can use to create augmented reality and virtual reality experiences. These components can be used in many different areas, from user interface elements to interaction management. For example, there are components such as buttons, sliders, menus, hand tracking, motion detection. MRTK has features that;
 - Interaction Components: MRTK provides a set of components that enable users to interact with the virtual world. These components support different interaction modes such as hand tracking, hand gesture detection, touch input, controller input.
 - Advanced Imaging and Visualization: MRTK provides visual feedback tools and elements to enable users to better understand the virtual world. For example, it can make users' interactions more meaningful with features such as distance indicators, eye-catching effects, and eye tracking.
 - Adaptive Interfaces: MRTK offers adaptive interfaces that can dynamically adapt to users' physical environments and the devices used. This allows adjusting interface elements based on screen size, resolution and other factors to optimize the user experience.
 - Rapid Prototyping and Development Tools: MRTK offers developers ready-made components and pre-configured features for rapid prototyping and development. This speeds up the application development process and saves time.
 - Movement and Positioning Tracking: MRTK provides tools to track users' movements and locations. This makes augmented reality and virtual reality experiences more immersive and realistic.

- Multi-Platform Support: MRTK has a structure that can run on different AR/VR devices and supports different game engines. This allows developers to deliver consistent experiences across various platforms.
 - Community Contribution and Development: MRTK is an open-source project and is constantly evolving with contributions from a large developer community. This ensures that new features are constantly added, bugs are fixed, and the platform is generally improved.
 - MRTK offers developers a powerful toolset to help them develop AR/VR applications and is used in many projects focused on Microsoft's Mixed Reality platform.
- Unlike the 2D environment, after adding the MRTK library, different approaches have been applied for actions such as clicking buttons and viewing images.
 - As in 2 dimensions, the information and images of all feature maps and filters taken from the Python application will be used in this application.
 - A hand menu is added for Lenet or Alexnet Model selection to be opened when the user goes anywhere.
 - As in 2 dimensions, the image objects of the feature maps that will be placed on the screen when the application is opened are created under the relevant layers. However, it is not active. A structure as in Table 3.1 is established and the relevant images will be active when the layer is opened from the model selected from the menu.
 - When the user puts on the glasses and starts the application, it will be in (0,0,0) position, so the entire environment is designed accordingly.
 - The distance of the layers to the user and the distance to the panel from which the input will be selected have been determined, and an effort has been made to ensure that the user can interact with the layers comfortably.
 - Since Alexnet has many layers, a structure surrounding the user has been created.
 - By giving volume to the layers, their depth is made visible.
 - You can click on the feature maps opened for Alexnet. In this way, the desired image is displayed in front of the user's eyes and he/she is given the opportunity to examine it more closely. In addition, it is set to manipulate the image to the desired size, rotate it and move it.

- The application has been packaged for the Hololens device and made ready for use.
- So the application steps are as follows:
 - The user selects the model type from the hand menu (**Figure 3.19**).
The selected model opens (**Figure 3.20**).

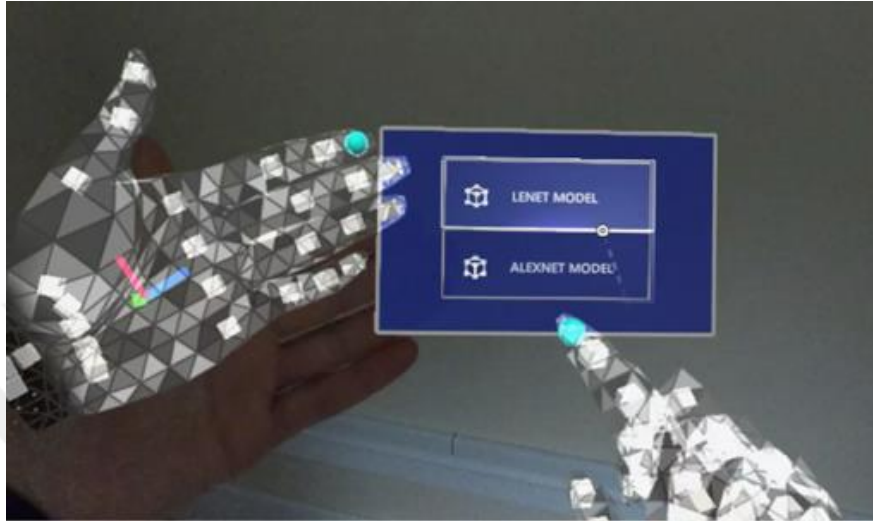


Figure 3.19: Hand menu that allows switching between models.

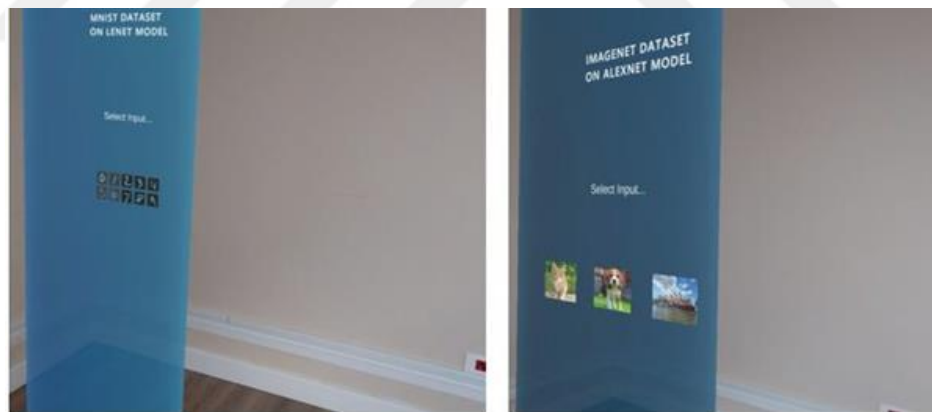


Figure 3.20: Formation of the selected model in front of the wall.

- The user selects an input from the test data. The feature maps of the selected image appear in layers (**Figure 3.21**).



Figure 3.21: Expressing the volumes of layers with cubes.

- The user opens and examines the feature maps by clicking on individual layers (**Figure 3.22, Figure 3.23**).

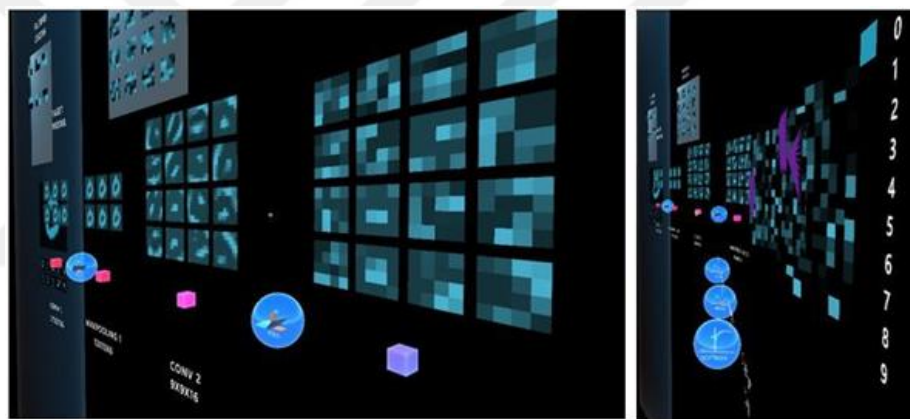


Figure 3.22: Examination of layers in the Lenet model.

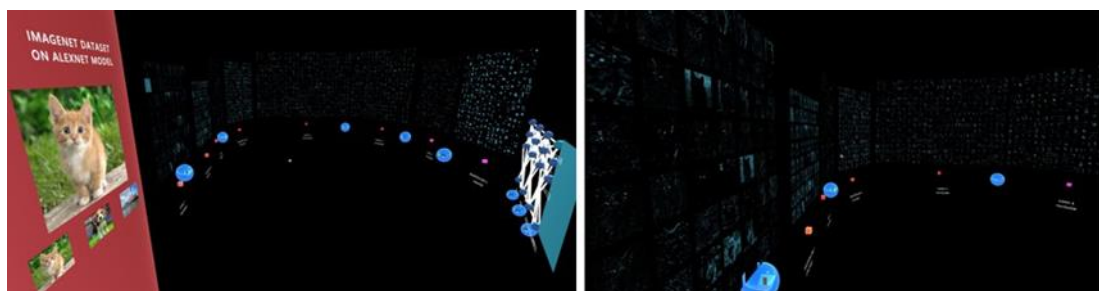


Figure 3.23: Examination of layers in the Alexnet model.

- Alexnet Model has the ability to examine the images closely because they are very small. The user clicks on the images and can even

examine the images on other layers by placing them side by side (Figure 3.24).

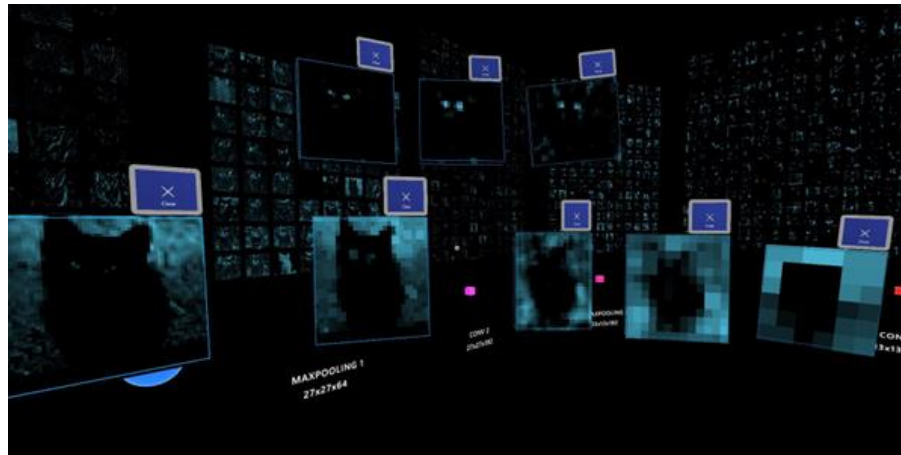


Figure 3.24: Close examination of images focusing on similar features.

- Bringing these images together, enlarging or reducing them, changing their rotation, in short, manipulation of the image can be achieved (Figure 3.25). For this, the features of the MRTK library are used.

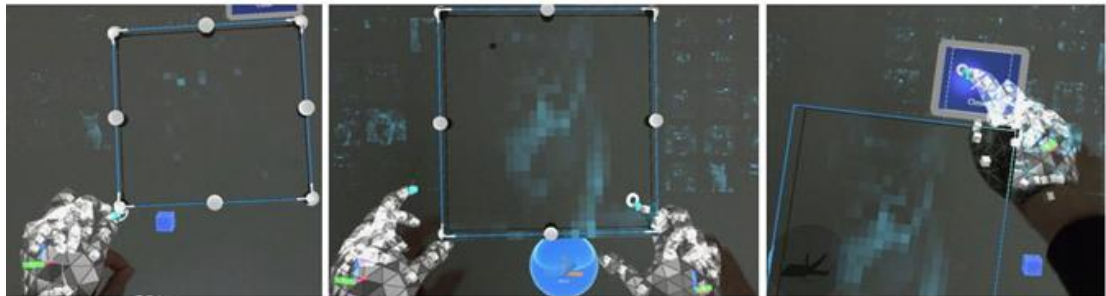


Figure 3.25: Object manipulation techniques used in the study.

- When selecting another entry, the images within the layers are closed and the user must click on the layer buttons again.
- It examines the prediction phase of this different input it selects with the updated feature maps.

The source code of the application can be accessed [65].

3.3. UX Study

3.3.1. Empathy and Problem Identification

During the user experience research phase, first of all, the pain points and gains of the users were determined regarding the study to be carried out. For this reason, research was conducted on how users use augmented reality in artificial intelligence systems, and the pain points thought to exist will turn into gains. An attempt was made to decide at what point in the artificial intelligence system it would be placed as a visualization application. Because, when looking at the artificial intelligence system, visualization can be provided at different stages as seen in **Figure 3.26**.

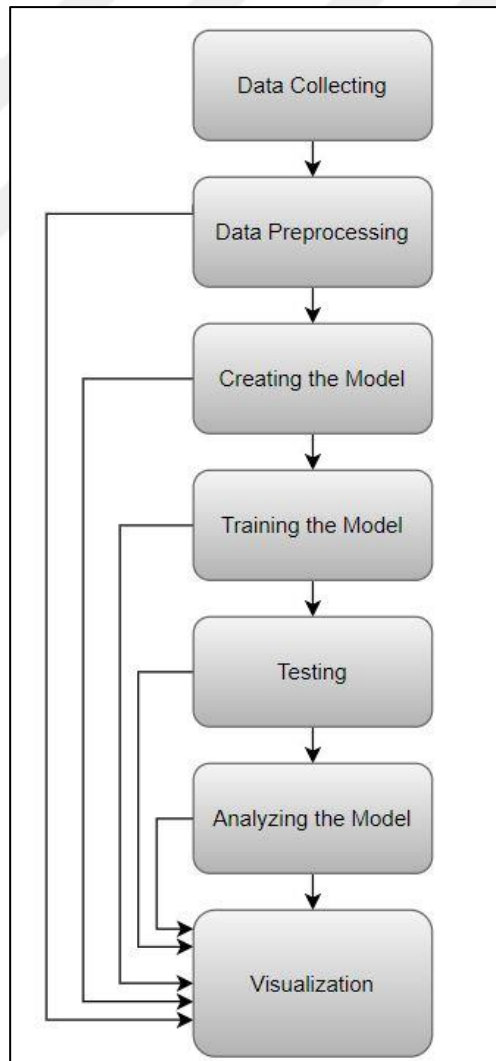


Figure 3.26: Stages where visualization can be used in the DL system.

For this purpose, a short survey was conducted on 5 users working in the artificial intelligence team in the company environment. Questions asked to users as in **Table 3.2**. The answers were as in **Table 3.3**. After examining the comments and thinking about the visualization in augmented reality; It was decided to conduct a study to relieve users' pain in visualizing complex models and seeing the learning process and internal structure of the model.

Table 3.2: Questions asked for user needs analysis.

1	Are there any applications you use for your parameter selections when creating your model? If so, are you satisfied with them?
2	What difficulties do you experience in collecting/preparing the data sets or resources you use in the training process?
3	What difficulties do you experience in interpreting the metrics you use in testing and validating your model?
4	What visualization tools do you use to understand the internal structure of the model and its learning process, and what difficulties are there in using these tools?
5	What techniques do you use to accurately analyze and interpret your model's predictions or results, and what challenges do these techniques present?

Table 3.3: User answers for analyzing user needs.

1	There are people who use Grid Search and AutoML tools for parameter selection. Apart from these, there are those who say that in some cases it is more flexible to manually adjust the parameters and observe the results.
2	The preprocessing steps required to improve the quality of datasets (data cleaning, handling missing data, feature engineering, etc.) were noted to be challenging. They were told that it might take time. It was said that it took time.
3	It was stated that the interpretation of the metrics used to evaluate the performance of the model can sometimes be complicated. Choosing the right metric can also be difficult. It was said that sometimes it may be necessary to use more than one metric.
4	There are people who use visualization tools such as tensorboard, keras-vis, or plotmodel to understand the internal structure of the model and the learning process. Challenges were expressed, such as the complexity of visualization tools and the need for certain technical knowledge.
5	There are those who use basic classification evaluation tools such as confusion matrix to analyze the model's predictions and results. It was stated that analysis and interpretation of predictions can be complicated depending on the complexity of the model.

- Creation of Personas

A persona is a fictional character created to represent the target audience of a product or service. This persona reflects the characteristics, needs, preferences and behaviors of real users. Factors such as demographic information, lifestyle, goals, needs, habits and motivations are generally taken into consideration when creating a persona. This information helps to better tailor the product or service to its target audience. It increases the success of the product and user satisfaction. In the study, 5 different personas were created as seen in **Figure 3.27** and **Figure 3.28**.

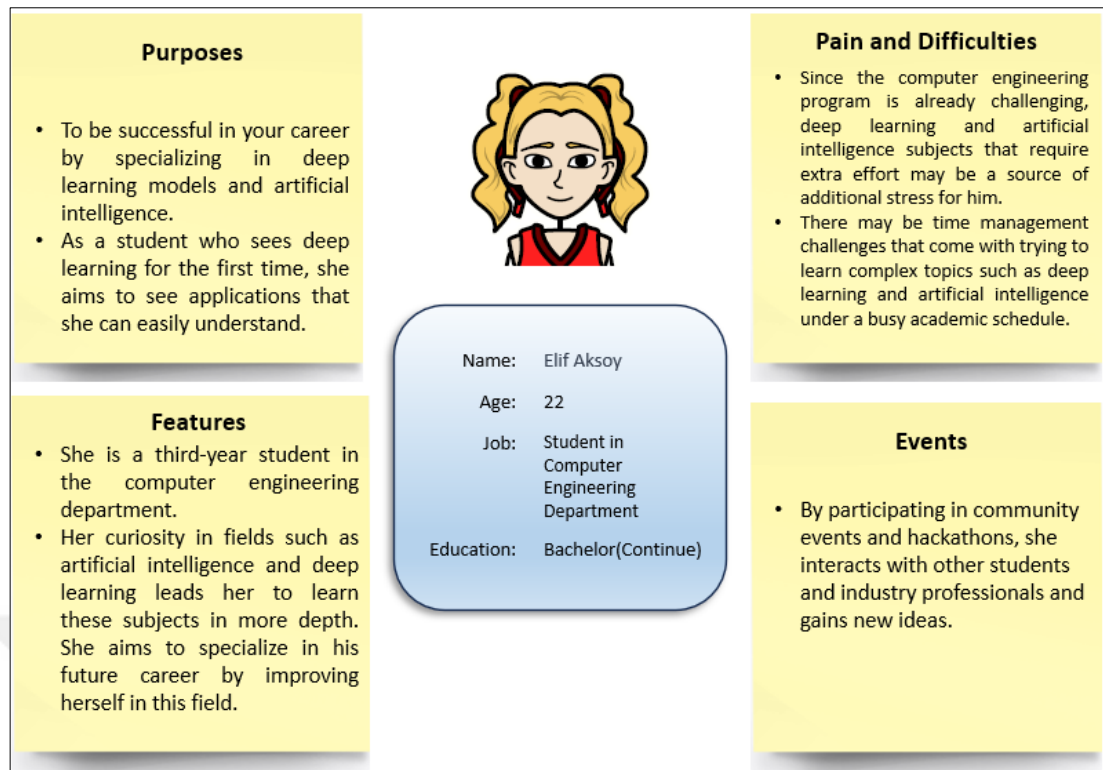


Figure 3.27: User personas are user profiles that represent the wants and needs of the target audience.

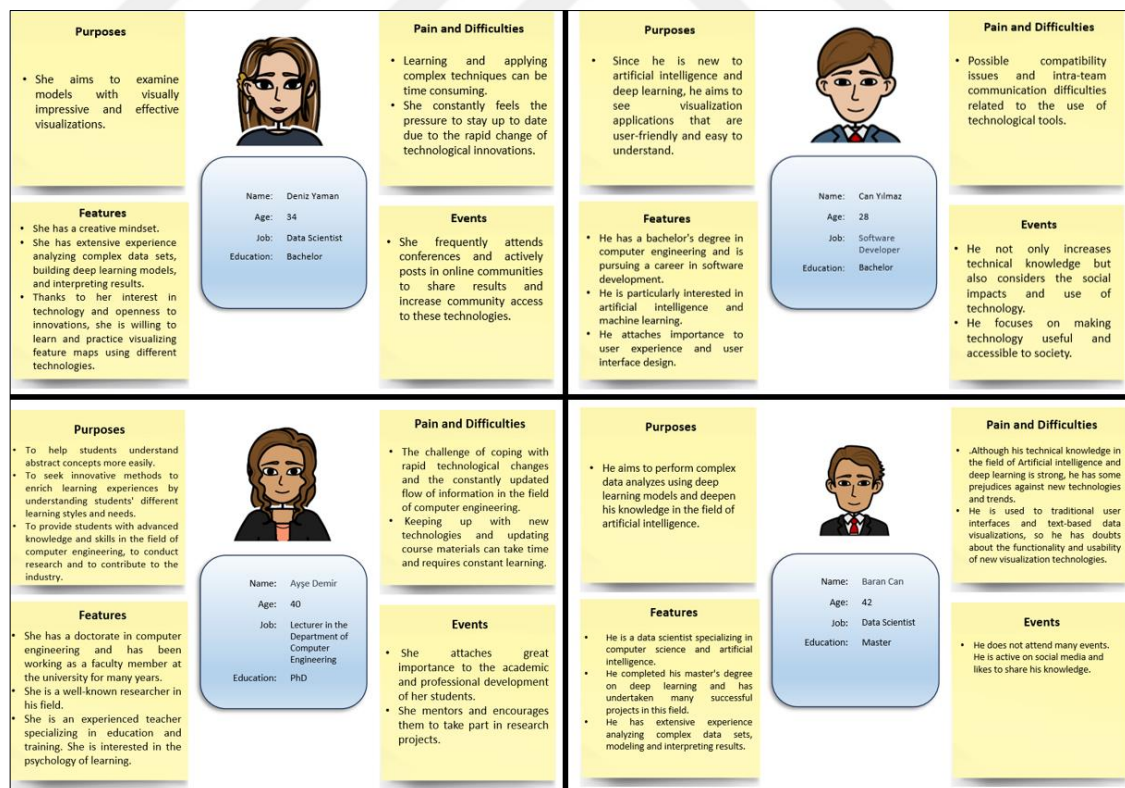


Figure 3.28: Personas created for the study.

3.3.2. Idea Development and Prototype Creation

After the step of knowing and understanding the user in the design-oriented thinking stages, an idea will be developed for the application and a decision will be made on what to do. For this purpose, feature visualization was used to understand what is happening in the prediction phase in the layers of deep learning models. In order to receive user feedback and see how the design works in practice, prototype development was started, the Lenet model was selected, a suitable prototype was designed and implemented on Unity. For the prototype design, a wireframe that can be used for 2D and AR applications was created, as seen in **Figure 3.29**. A wireframe is a simple, linear outline that shows the basic structure of a page or application. It doesn't focus on details like color, font, or visual elements; instead, it emphasizes the layout, layout, and functionality of the user interface. A wireframe is usually created using boxes, lines and simple text. It is used in the early stages of design and developed into detailed designs in later stages. This is an important step to ensure that the design requirements and user needs are properly understood. For this purpose, Figma application, which is a cloud-based design tool and supports many design processes from user interface design to prototyping, was used [66]. Figma also has support for the Hololens-2 interface design.

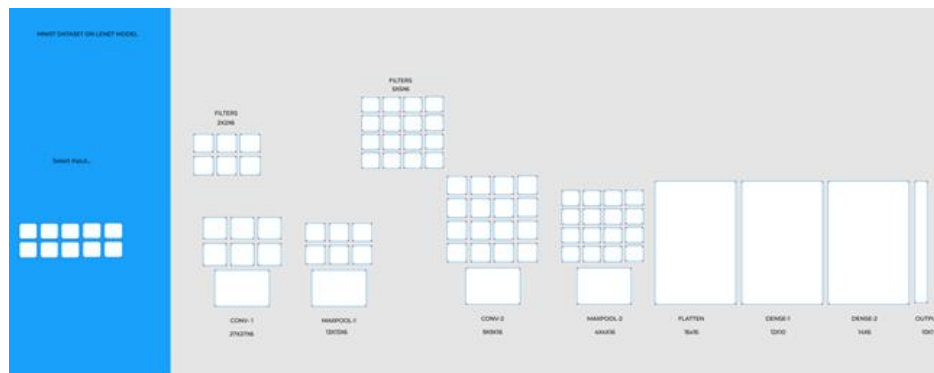


Figure 3.29: Creating a Wireframe for the prototype of a 2D application.

The designed prototype was transformed into 2 different applications for 2D and 3D environments, as shown in **Figure 3.30**, by going through the stages described in Sections 3.1 and 3.2. Prepared for users to test.

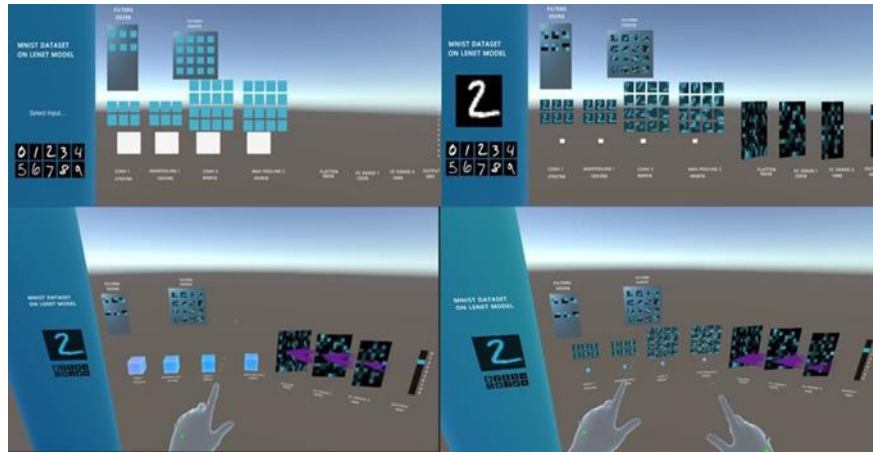


Figure 3.30: Application created after prototype.

3.3.3. User Testing

The user testing process involves obtaining valuable information by observing how the designed product or service is used by real users, recording the resulting data, and analyzing this data. This process allows different users to perform specific tasks using the product. These tasks represent the specified purposes and usage scenarios of the product. Users provide feedback about the difficulties they encounter while completing tasks, points they have difficulty understanding, or issues they feel positive or negative about the experience the product offers. This feedback provides important clues for making improvements to the product's design and user experience. User testing and usability testing are different from each other. User testing is usually implemented earlier in the product development process and aims to evaluate users' experiences when using the product in real-world scenarios. This testing is used to understand how users use the product in real use cases and what problems they encounter. User testing is generally observation-based. Users are monitored as they perform specific tasks and observations are recorded. Additionally, users can be interviewed and feedback can be obtained. Usability testing is usually applied at later stages of the product development process and aims to evaluate how well the product meets certain usability criteria. This testing is used to measure specific metrics related to product usage and systematically evaluate users' interaction with the product. Usability testing is usually performed on a specific test scenario. Users are observed using the product to complete specific tasks

and their performance is measured. Usability testing is often evaluated using specific criteria or standards. User testing was conducted with users with the features as seen in **Table 3.4** and **Table 3.5**.

Table 3.4: User interests.

	Groups	Frequency	Percentage (%)
Title	Software Engineer Interested in Data Science	10	50
	Software Engineer Interested in Augmented Reality	4	20
	Software Engineer Interested in Augmented Reality and Data Science	2	10
	Software Engineer Not Interested in Augmented Reality and Data Science	4	20
	Total	20	100

Table 3.5: User experiences.

	Groups	Frequency	Percentage (%)
Experience	1-5 year	10	50
	6-10 year	10	50
	Total	20	100

The application was first run as a desktop application and later on the Hololens-2, a mixed reality headset. The applications were tested individually with 20 people. Users were given tasks and given preliminary information on how to do them. Tasks;

1. Task 1: Open the application on the PC and select the test data. Which part of the test data is focused on in the 2nd convolution layer?
2. Task 2: Which convolution filter detects certain features in the data more effectively on PC?
3. Task 3: Open the application on the Hololens 2 and select the test data. Which part of the test data is focused on in the 2nd convolution layer?

4. Task 4: Which convolution filter detects certain features in the data more effectively on Hololens 2.

Table 3.6: Questions asked after user testing.

What is your feedback about 2 uses?
Which features have been the most useful or most complex for you?
What were the difficulties you experienced while completing the tasks?
Did you find the content and information provided sufficient?

General questions were asked to users who completed their tasks as shown in **Table 3.6**. As a result of the responses received, it was determined that users generally had initial difficulties in using Hololens. It was observed that users who are not interested in augmented reality had difficulty in clicking, especially. It seemed that everyone was willing to use the new system.

Looking at the answers received from the questions, it was seen that the users gave answers that were open to improvement in the interfaces in the 3D environment.

They stated that being interactive in a 3D environment could make it easier to understand more complex networks. In this sense, they added that they want to use augmented reality interfaces.

In the continuation of the study, UX studies were continued with the aim of developing a system that will enable the visualization of the model structure based on this foundation. At this stage, users are presented with a structure where they can create ideas. UX studies were advanced by adding the Alexnet model. After this stage, the developed application was tested with more users. Usability was tested by determining the usability metric. The survey has been prepared. After the assigned tasks, the user filled out a survey and usability, including satisfaction, was measured.

3.3.4. Determine Usability Metric

User research should identify the most important statistical information and tools needed to measure user experience, providing guidelines on the practical aspects of quantitative analysis. Quantitative analysis is an important tool for measuring and understanding user experience. In this study, the method called "Single Usability Metric" was used for usability. It is a concept used to collect the usability of a product or experience under a single criterion. It is often used to simplify complex availability data and simplify the decision-making process. It was decided that task completion rates, number of errors, user satisfaction and time taken to complete the task would be appropriate to combine this information under a single metric to understand and compare usability more easily. The SUM score summarizes the usability performance of an interface design in a single number. This allows to compare different designs quickly and quantitatively. One of the advantages of SUM is that it summarizes the user experience in a single metric, helping to make design decisions more objectively. Additionally, by using it in conjunction with qualitative data obtained through iterative usability testing, the SUM score can provide a more comprehensive assessment of user experience. These tests provide qualitative data, such as how users perceive interface designs, how easy they are to use, and what their overall experience is like. This data is used in conjunction with the SUM score to enable the design to be evaluated from both quantitative and qualitative perspectives so that necessary improvements can be identified to better suit the design's user needs. For example, a single numerical value can be obtained by taking a weighted average of different metrics such as task completion rates, number of errors, and user satisfaction scores. This will be useful to measure user experience across several different dimensions and present them under a single metric.

For the SUM technique, task and survey planning was made for the users. Google Forms application was used for the survey (**Figure 3.32**). Users were given access to the survey page with the help of a QR code to answer the questions (**Figure 3.31**).

Preliminary information was given by writing the tasks (**Table 3.7** and **Table 3.8**). First, it was questioned whether the users would like to participate in the survey and whether they would like to take photographs for use. At the end of the task, questions were asked first about the tasks and then about satisfaction. It was decided to

calculate satisfaction with the ASQ technique given in Figure 2.3. The questions have been prepared. At the end of each task, users were asked some questions.



Figure 3.31: The survey was accessed via QR code.

Visualization of Deep Learning Models with 2D Interfaces and Augmented Reality Interfaces		Personal Information														
It is a survey of a thesis study to investigate user perception on 2D interfaces and augmented reality interfaces in deep learning models. The application includes the visualization of feature maps of the Lenet and Alexnet Model. The application is intended to display pixel information of images in layers. While a laptop will be used for 2 dimensions, a hololens-2 device will be used for augmented reality. - The understandability of which layer and which feature is active in the learning phase will be measured. - Answers will be received in 2 dimensions and augmented reality. - The usability of 2 applications will be measured.		What is your employment status? <table border="1"> <thead> <tr> <th></th> <th>Bachelor</th> <th>Master</th> <th>PhD</th> </tr> </thead> <tbody> <tr> <td>I am a student.</td> <td>☐</td> <td>☐</td> <td>☐</td> </tr> <tr> <td>I am working. (Education)</td> <td>☐</td> <td>☐</td> <td>☐</td> </tr> </tbody> </table>				Bachelor	Master	PhD	I am a student.	☐	☐	☐	I am working. (Education)	☐	☐	☐
	Bachelor	Master	PhD													
I am a student.	☐	☐	☐													
I am working. (Education)	☐	☐	☐													
Date gg.aa.yyyy		If you are working, please specify your profession.														
Do you want to participate in the study? ☐ Yes ☐ No		If you are working, please indicate your years of experience.														
Do you allow photographs to be taken during the study to be used within the scope of the thesis? ☐ Yes ☐ No		Are you interested in Deep Learning or Augmented Reality? ☐ I'm interested in Deep Learning. ☐ I'm interested in Augmented Reality. ☐ I'm interested in both. ☐ I'm not interested in either of them.														
		Which app did you use first? ☐ 2D Interfaces ☐ Augmented Reality Interfaces														

Figure 3.32: The survey used in the study.

Table 3.7: Tasks for 2D Interfaces on laptop application.

TASK – 1	TASK - 2
1. Select the Lenet model from the application opened on the laptop. 2. Select the input you want to view. 3. Click on the buttons to access the feature maps in the layers. 4. Answer the questions below in the layer information you clicked on.	1. Select the Alexnet model from the application opened on the laptop. 2. Select the input you want to view. 3. Click on the buttons to access the feature maps in the layers. 4. Answer the questions below in the layer information you clicked on.
Can you infer which feature the layer detects?	Can you infer which feature the layer detects?
After examining the feature maps, were you able to understand how the model discriminates certain features?	After examining the feature maps, were you able to understand how the model discriminates certain features?
Do the visual feature maps clearly show you which features the deep learning model focuses on during training?	Do the visual feature maps clearly show you which features the deep learning model focuses on during training?
Can you understand the relationships between visual feature maps? For example, how do you think a connection is made between a particular attribute of an object and another attribute? What do the relationships between attributes tell you about the model's thought processes?	Can you understand the relationships between visual feature maps? For example, how do you think a connection is made between a particular attribute of an object and another attribute? What do the relationships between attributes tell you about the model's thought processes?
Were you able to understand the process of creating feature maps of filters?	

Table 3.8 : Tasks for AR Interfaces on hololens-2 application.

TASK – 1	TASK - 2
1. Look at your hand in the opened application. See the menu. 2. In the menu that appears, select the Lenet model. 3. Select the input you want to view. Make sure that the line coming out of your finger is on the input you will select. After selecting, the image will appear at the top. 4. Click on the cubes to access the feature maps in the layers. 5. Answer the questions below in the layer information you clicked on.	1. Look at your hand in the opened application. See the menu. 2. In the menu that appears, select the Alexnet model. 3. Select the input you want to view. Make sure that the line coming out of your finger is on the input you will select. After selecting, the image will appear at the top. 4. Click on the cubes to access the feature maps in the layers. 5. Click to take a closer look at the feature maps that appear. 6. Enlarge or reduce the image in front of you. Carry it. 7. Answer the questions below in the layer information you clicked on.
Can you infer which feature the layer detects?	Can you infer which feature the layer detects?
After examining the feature maps, were you able to understand how the model discriminates certain features?	After examining the feature maps, were you able to understand how the model discriminates certain features?
Do the visual feature maps clearly show you which features the deep learning model focuses on during training?	Do the visual feature maps clearly show you which features the deep learning model focuses on during training?
Can you understand the relationships between visual feature maps? For example, how do you think a connection is made between a particular attribute of an object and another attribute? What do the relationships between attributes tell you about the model's thought processes?	Can you understand the relationships between visual feature maps? For example, how do you think a connection is made between a particular attribute of an object and another attribute? What do the relationships between attributes tell you about the model's thought processes?
Were you able to understand the process of creating feature maps of filters?	

4. EXPERIMENT AND RESULTS

The issue to be investigated within the scope of this study is what kind of effect AR interfaces will have on DL models. Based on this, a test was conducted to test the reactions of users in existing systems and the reactions of users when AR was added. Users are selected GTU Computer Engineering Department undergraduate, graduate and PhD students and Havelsan Inc. from Artificial Intelligence Team and Augmented and Virtual Reality Teams. A/B testing has been applied to this system, which has the same interface.

4.1. Creating User Group

First of all, the dependent and independent variables are determined here. The independent variable was the interface type based on the platform used. This variable is a two-level variable; interfaces on the two-dimensional computer screen and augmented reality interfaces on the Hololens-2 device. The dependent variable is user experience. This variable was evaluated with measurements such as whether users could complete the given tasks, duration of use, and degree of satisfaction.

Users are divided into two groups: controlled and experimental groups. One group, those who tested the application in a two-dimensional computer environment, was the control group, while the other group, those who tested the application in a three-dimensional environment with the Hololens device, was the experimental group. Two versions of the application were applied separately to the control and experimental groups. While the group with the two-dimensional version used the application on the laptop screen, the group with the augmented reality version experienced the application using the Hololens-2 device.

4.2. Experiment

During testing, users were asked to primarily use a single system. The user using a single system was asked to complete a survey only about that application without being shown the other system. Among the questions, the first question was about which system he used. After performing the test, the other system that he did not use

was shown and then a survey of that system was made. In addition to A-B testing, the aim is to measure how the same user is affected when he sees the other system.

Efforts have been made to ensure equal distribution of users according to their experience, educational level and knowledge on this subject. As seen in Figure 4.1, two systems were tested with users.

The system was tested with 51 people. For equal distribution, 25 people were set as the control group and 25 people as the experimental group. The answers to the questions asked to the users whether they are students or not, if they are working, their profession and years of experience, their field of interest, and on which device they used the test first, were taken as shown in **Table 4.1**, **Table 4.2**, **Figure 4.1** and **Figure 4.2**.

Table 4.1: Occupational distribution of users.

Occupation	Frequency
Computer/Software/R&D/Artificial Intelligence/Data Eng.	13
Academician	3
XR Eng.	2
Mathematics Eng.	1

Table 4.2: Educational status distribution of users.

Program	Frequency
Undergraduate Class 1-2	17
Undergraduate Class 4	13
Bachelor's degree	6
Master (Student)	8
Master's Degree (Graduate)	3
Doctorate	3

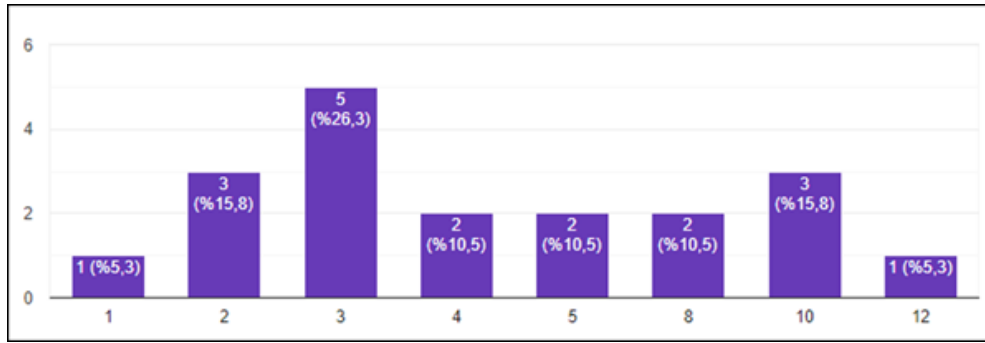


Figure 4.1: Years of user's experience.

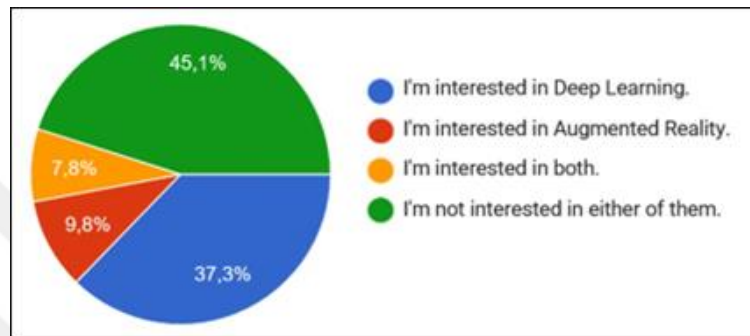


Figure 4.2: Users' interests.

4.3. Data Collection

Users are given the tasks shown in the **Table 3.7** And **Table 3.8** for both systems. The systems were experienced by the users as shown in **Figure 4.3** and **Figure 4.4**. Feedback from users was received at the end of the tasks using the Google Form application. This application provides the answers received as an .xlsx file. These answers were taken and categorized as in **Figure 4.5**.

At the end of each version application, whether the users were able to complete the given task; It was determined by whether the model could understand how the feature maps formed in the process of reaching the result progressed. Satisfaction; It was measured by surveying users. Duration of use was noted at the end of the task.

In the evaluation phase of these results, the SUM technique determined as described in section 3.3.4 was applied for usability testing.



Figure 4.3: Testing 2d system with users.



Figure 4.4: Testing AR system with users.

Undergraduate Class 1./2.							
1	1/18/24	Yes	Yes	Bachelor			I'm interested in Augmented Re 2D Interfaces
2	1/18/24	Yes	Yes	Bachelor			I'm interested in Augmented Re Augmented Reality Interfaces
3	1/23/24	Yes	Yes	Bachelor			I'm interested in both. 2D Interfaces
4	1/23/24	Yes	No	Bachelor			I'm not interested in either of the 2D Interfaces
5	1/23/24	Yes	Yes	Bachelor			I'm not interested in either of the 2D Interfaces
6	1/23/24	Yes	Yes	Bachelor			I'm not interested in either of the 2D Interfaces
7	1/23/24	Yes	Yes	Bachelor			I'm not interested in either of the 2D Interfaces
8	1/23/24	Yes	Yes	Bachelor			I'm not interested in either of the Augmented Reality Interfaces
9	1/23/24	Yes	Yes	Bachelor			I'm not interested in either of the Augmented Reality Interfaces
10	1/23/24	Yes	Yes	Bachelor			I'm not interested in either of the Augmented Reality Interfaces
11	1/23/24	Yes	Yes	Bachelor			I'm not interested in either of the Augmented Reality Interfaces
12	1/23/24	Yes	Yes	Bachelor			I'm not interested in either of the Augmented Reality Interfaces
13	1/24/24	Yes	Yes	Bachelor			I'm not interested in either of the 2D Interfaces
14	1/24/24	Yes	Yes	Bachelor			I'm interested in Deep Learning 2D Interfaces
15	1/24/24	Yes	Yes	Bachelor			I'm not interested in either of the Augmented Reality Interfaces
16	1/24/24	Yes	Yes	Bachelor			I'm interested in Deep Learning Augmented Reality Interfaces
17	1/24/24	Yes	Yes	Bachelor			I'm not interested in either of the Augmented Reality Interfaces
Undergraduate Class 4.							
1	1/16/24	Yes	Yes	Bachelor			I'm interested in Deep Learning 2D Interfaces
2	1/21/24	Yes	Yes	Bachelor			I'm interested in Deep Learning 2D Interfaces
3	1/21/24	Yes	Yes	Bachelor			I'm not interested in either of the 2D Interfaces
4	1/21/24	Yes	Yes	Bachelor			I'm not interested in either of the 2D Interfaces
5	1/21/24	Yes	Yes	Bachelor			I'm interested in Deep Learning 2D Interfaces
6	1/21/24	Yes	Yes	Bachelor			I'm not interested in either of the 2D Interfaces
7	1/21/24	Yes	Yes	Bachelor			I'm interested in Deep Learning 2D Interfaces
8	1/21/24	Yes	Yes	Bachelor			I'm not interested in either of the Augmented Reality Interfaces
9	1/21/24	Yes	Yes	Bachelor			I'm not interested in either of the Augmented Reality Interfaces
10	1/21/24	Yes	No	Bachelor			I'm not interested in either of the Augmented Reality Interfaces
11	1/21/24	Yes	Yes	Bachelor			I'm not interested in either of the Augmented Reality Interfaces
12	1/21/24	Yes	No	Bachelor			I'm interested in Augmented Re Augmented Reality Interfaces
13	1/21/24	Yes	Yes	Bachelor			I'm interested in Deep Learning Augmented Reality Interfaces
Master							
1	1/16/24	Yes	Yes	Master	AI Engineer	1	I'm interested in Deep Learning 2D Interfaces
2	1/16/24	Yes	Yes	Master			I'm interested in Deep Learning 2D Interfaces
3	1/16/24	Yes	Yes	Master	Computer Eng	12	I'm interested in both. 2D Interfaces
4	1/16/24	Yes	Yes	Master	Software Engi	10	I'm interested in Deep Learning 2D Interfaces
5	1/18/24	Yes	Yes	Master	Computer Eng	2	I'm not interested in either of the 2D Interfaces
6	1/18/24	Yes	Yes	Master	R&D Engineer	2	I'm interested in Deep Learning Augmented Reality Interfaces
7	1/18/24	Yes	Yes	Master	Computer Eng	4	I'm interested in Deep Learning Augmented Reality Interfaces
8	1/24/24	Yes	Yes	Master	Computer Eng	3	I'm interested in Deep Learning Augmented Reality Interfaces
9	1/24/24	Yes	Yes	Master	Software Engi	5	I'm interested in Deep Learning Augmented Reality Interfaces
10	1/24/24	Yes	Yes	Master	Software Engi	3	I'm interested in Deep Learning Augmented Reality Interfaces
11	1/25/24	Yes	Yes	Master	Software Engi	8	I'm not interested in either of the Augmented Reality Interfaces
PhD							
1	1/22/24	Yes	Yes	PhD	Academician	8	I'm interested in both. Augmented Reality Interfaces
2	1/24/24	Yes	Yes	PhD			I'm interested in Deep Learning 2D Interfaces
3	1/24/24	Yes	Yes	PhD	Research Ass	3	I'm interested in Deep Learning 2D Interfaces
Bachelor							
1	1/24/24	Yes	Yes	Bachelor	Software Engi	2	I'm interested in Deep Learning 2D Interfaces
2	1/24/24	Yes	Yes	Bachelor	Data Engineer	4	I'm interested in Deep Learning Augmented Reality Interfaces
3	1/24/24	Yes	Yes	Bachelor	Software Engi	3	I'm interested in Deep Learning Augmented Reality Interfaces
AR							
1	1/24/24	Yes	Yes	Bachelor	XR Engineer	5	I'm interested in Augmented Re Augmented Reality Interfaces
2	1/24/24	Yes	Yes	Bachelor	XR Engineer	3	I'm interested in Augmented Re 2D Interfaces
Maths							
1	1/17/24	Yes	Yes	Bachelor	Maths Engine	10	I'm not interested in either of the 2D Interfaces
2	1/17/24	Yes	Yes	Bachelor	Maths Engine	10	I'm not interested in either of the Augmented Reality Interfaces

Figure 4.5: Classification of user answers.

4.4. Results

In this section, the answers given by the users, the completion status of the tasks and the time taken to complete the tasks were analyzed. The SUM technique expressed in **Figure 4.6** was used. Users who were given 2 different tasks on 2 different platforms answered the survey questions by performing the tasks.

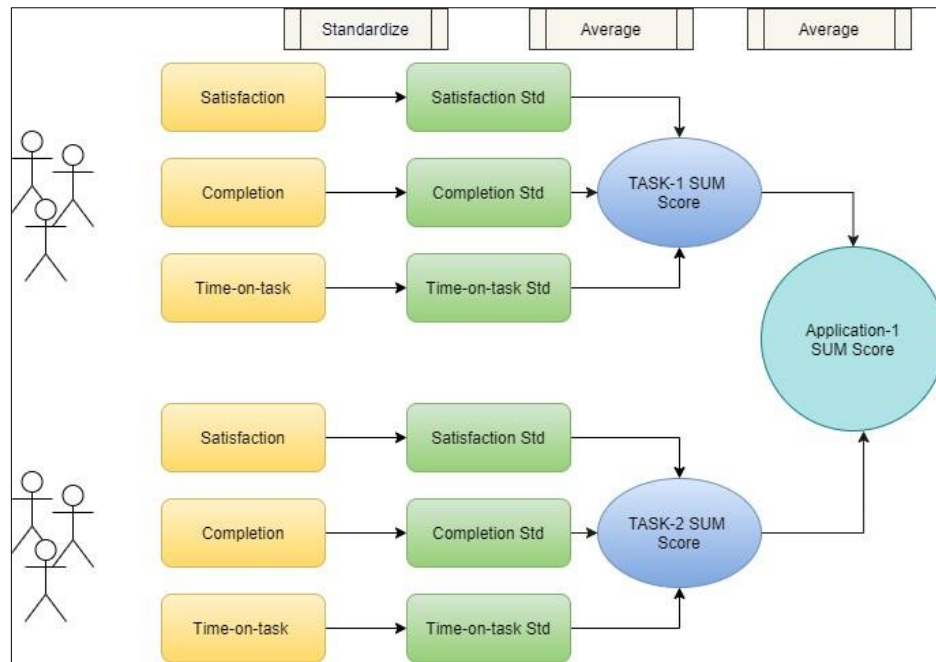


Figure 4.6: SUM usage pattern.

4.4.1. Satisfaction Measurement

Three of the questions asked to users are ASQ that measure satisfaction, and users were asked to evaluate satisfaction between 1-7:

- Overall, I am pleased with the ease of completing the mission in this scenario.
- Overall, I'm happy with the time it took to complete the task in this scenario.
- Overall, I am satisfied with the support information (online help, messages, documentation) provided when completing the task.

According to the results of this survey, the average was taken, and the answers were calculated between 1-5 to give it to the SUM calculation tool. Thus, the result was as seen in **Figure 4.7**.

	task-1	task-2	task-3	task-4		task-3	task-4	task-1	task-2
id	satisfaction	satisfaction	satisfaction	satisfaction	id	satisfaction	satisfaction	satisfaction	satisfaction
11	3.571428571	2.857142857	3.80952381	3.571428571	12	3.095238095	3.333333333	4.761904762	5
34	4.523809524	4.285714286	5	4.523809524	37	4.047619048	4.285714286	4.523809524	4.523809524
36	4.523809524	4.523809524	4.047619048	4.047619048	38	5	4.761904762	4.523809524	4.523809524
26	5	5	5	4.761904762	39	5	4.761904762	4.523809524	3.80952381
40	5	4.285714286	5	4.761904762	44	5	5	5	5
41	4.761904762	4.285714286	4.523809524	4.523809524	45	5	4.761904762	5	5
42	4.761904762	4.523809524	4.523809524	4.523809524	46	4.761904762	4.761904762	4.523809524	4.523809524
43	5	4.523809524	4.761904762	4.523809524	47	4.285714286	4.285714286	3.80952381	3.80952381
5	5	4.285714286	4.761904762	4.761904762	48	5	5	4.761904762	4.761904762
13	5	4.523809524	5	4.761904762	19	5	5	4.761904762	4.761904762
14	4.761904762	4.761904762	5	4.761904762	20	4.047619048	4.047619048	4.047619048	4.047619048
15	4.761904762	4.523809524	4.761904762	4.761904762	21	5	4.523809524	5	5
16	4.523809524	4.761904762	5	4.285714286	22	4.285714286	4.523809524	5	4.285714286
17	5	5	4.761904762	4.761904762	23	4.523809524	5	5	5
18	4.761904762	4.523809524	4.761904762	4.761904762	24	4.761904762	5	5	4.761904762
1	5	4.285714286	4.047619048	4.285714286	9	4.761904762	4.761904762	3.571428571	3.571428571
2	3.333333333	2.857142857	3.095238095	3.333333333	10	4.523809524	4.523809524	5	5
3	5	4.047619048	4.523809524	4.523809524	27	5	5	5	4.285714286
4	3.80952381	3.80952381	3.80952381	3.80952381	49	4.285714286	4.523809524	3.571428571	3.571428571
8	3.333333333	3.333333333	4.047619048	4.047619048	50	4.285714286	4.285714286	3.80952381	3.80952381
33	3.571428571	3.571428571	4.285714286	4.285714286	32	3.571428571	3.571428571	4.761904762	3.80952381
35	4.523809524	4.285714286	3.333333333	3.333333333	25	5	5	2.142857143	2.142857143
28	5	4.523809524	5	5	30	4.047619048	4.047619048	3.80952381	3.80952381
51	3.571428571	3.571428571	4.285714286	4.285714286	31	5	5	5	4.523809524
6	4.285714286	4.285714286	4.285714286	4.285714286	29	4.761904762	4.761904762	5	4.047619048
	4.495238095	4.20952381	4.457142857	4.371428571		4.561904762	4.580952381	4.476190476	4.295238095

Figure 4.7: Averages of answers for ASQ and its equivalent for 5.

4.4.2. Task Completion

The tasks that users were asked to do before starting the test were explained. Afterwards, the test was started. During the test, he was asked to provide information about the feature maps and questions about the feature he focused on. The completion status of users who could not complete this task was taken as 0, while the value was taken as 1 for users who could respond. The results are seen in **Figure 4.8**.

	task-1	task-2	task-3	task-4		task-3	task-4	task-1	task-2
id	completion	completion	completion	completion	id	completion	completion	completion	completion
11	0	0	0	0	12	0	0	1	1
34	1	1	1	1	37	0	1	1	1
36	1	1	0	1	38	0	0	1	1
26	0	0	1	1	39	0	0	1	1
40	1	1	1	1	44	1	1	1	1
41	1	1	1	1	45	1	1	1	1
42	1	1	1	1	46	1	1	1	1
43	1	1	1	1	47	1	1	1	1
5	1	0	1	1	48	1	1	1	1
13	0	0	0	1	19	0	0	1	1
14	0	0	1	1	20	0	0	0	0
15	0	1	0	1	21	0	0	0	0
16	0	0	1	1	22	1	1	1	1
17	1	1	1	1	23	1	1	1	1
18	1	1	1	1	24	1	1	1	1
1	0	1	1	1	9	0	0	1	1
2	1	1	1	1	10	1	1	1	1
3	1	0	0	0	27	0	0	1	1
4	0	0	1	0	49	0	0	1	1
8	0	0	1	1	50	1	1	1	1
33	1	1	1	1	32	0	0	1	1
35	1	1	0	1	25	0	0	1	1
28	1	1	1	1	30	0	0	1	1
51	1	1	1	1	31	1	1	1	1
6	1	1	1	1	29	0	0	0	1

Figure 4.8: Task completion status.

4.4.3. Time Measurement

During the tests, time was calculated for each task. SUM in seconds is given to the calculation tool. Since Alexnet is more complex, it is normal for users to spend more time on Task-2 and Task-4. In addition, users who saw the Hololens-2 device for the first time wanted to use it independently of the task and extend the experience. Comments were made on the attributes and time measurement was completed at the end of the task. Afterwards, the time spent by the user is not taken into account. Although they were the first to use the system, there were also those who used it quickly. The results are seen in **Figure 4.9**.

	task-1	task-2	task-3	task-4		task-3	task-4	task-1	task-2
id	time	time	time	time	id	time	time	time	time
11	240	360	540	660	12	120	180	180	240
34	300	360	240	360	37	300	480	240	360
36	300	360	300	420	38	240	420	180	300
26	180	240	240	480	39	240	300	180	180
40	360	600	420	480	44	420	540	300	360
41	300	360	300	480	45	300	300	180	240
42	240	360	360	720	46	240	480	180	300
43	240	300	360	480	47	180	240	180	300
5	240	360	180	300	48	240	360	120	300
13	180	300	300	480	19	180	300	120	240
14	120	180	180	300	20	240	360	180	240
15	180	240	180	300	21	240	360	180	300
16	120	240	420	600	22	120	180	120	120
17	120	300	180	240	23	180	180	180	240
18	180	240	180	300	24	180	240	120	180
1	120	180	240	300	9	180	240	180	300
2	120	240	180	300	10	240	300	120	240
3	120	360	180	180	27	120	120	120	180
4	180	300	120	240	49	120	120	180	240
8	240	300	240	480	50	180	240	240	300
33	180	240	360	480	32	180	240	240	240
35	180	300	240	300	25	180	240	120	180
28	180	240	240	300	30	120	240	180	300
51	120	180	120	240	31	240	300	240	300
6	240	360	480	720	29	120	120	180	300

Figure 4.9: Duration of tasks (s).

Figure 4.10, Figure 4.11, Figure 4.12 and Figure 4.13 shows the result obtained by giving satisfaction, completion and duration values to the SUM calculation tool of the users who completed Task-1, Task-2, Task-3 and Task-4.

After the users completed the first application and answered the survey questions, the other application they did not use was tested. **Figure 4.14, Figure 4.15, Figure 4.16 and Figure 4.17** shows the result obtained by giving satisfaction, completion and duration values to the SUM calculation tool of the users who completed Task-1, Task-2, Task-3 and Task-4.

Copyright © 2011 Measuring Usability LLC <http://www.measuringusability.com>

Summary				
	SUM	Completion	Sat	Time
High	87.7%	77.7%	88.1%	97.2%
Avg.	79.4%	62.6%	78.5%	93.2%
Low	68.5%	47.6%	68.9%	89.1%

Task Raw Data					
User	Satisfaction	Completion	Times	Standardized Av	% Avg
1	3.571428571	0	240	-0.38	35%
2	4.523809524	1	300	0.63	73%
3	4.523809524	1	300	0.63	73%
4	5	0	180	0.70	76%
5	5	1	360	0.72	76%
6	4.761904762	1	300	0.76	78%
7	4.761904762	1	240	0.98	84%
8	5	1	240	1.11	87%
9	5	1	240	1.11	87%
10	5	0	180	0.70	76%
11	4.761904762	0	120	0.96	83%
12	4.761904762	0	180	0.56	71%
13	4.523809524	0	120	0.82	79%
14	5	1	120	1.78	96%
15	4.761904762	1	180	1.25	89%
16	5	0	120	1.09	86%
17	3.333333333	1	120	0.84	80%
18	5	1	120	1.78	96%
19	3.80952381	0	180	0.03	51%
20	3.333333333	0	240	-0.52	30%
21	3.571428571	1	180	0.58	72%
22	4.523809524	1	180		1.12 87%
23	5	1	180		1.39 92%
24	3.571428571	1	120		0.98 84%
25	4.285714286	1	240		0.71 76%

Figure 4.10: SUM score calculation based on user answers for Task-1.

Copyright © 2011 Measuring Usability LLC <http://www.measuringusability.com>

Summary				
	SUM	Completion	Sat	Time
High	83.8%	77.7%	75.9%	97.7%
Avg.	74.4%	62.6%	63.4%	94.0%
Low	62.9%	47.6%	50.8%	90.4%

Task Raw Data					
User	Satisfaction	Completion	Times	Standardized Av	% Avg
1	2.857142857	0	360	-0.81	21%
2	4.285714286	1	360	0.71	76%
3	4.523809524	1	360	0.84	80%
4	5	0	240	0.92	82%
5	4.285714286	1	600	0.09	54%
6	4.285714286	1	360	0.71	76%
7	4.523809524	1	360	0.84	80%
8	4.523809524	1	300	1.06	86%
9	4.285714286	0	360	0.02	51%
10	4.523809524	0	300	0.37	65%
11	4.761904762	0	180	1.13	87%
12	4.523809524	1	240	1.33	91%
13	4.761904762	0	240	0.78	78%
14	5	1	300	1.34	91%
15	4.523809524	1	240	1.33	91%
16	4.285714286	1	180	1.54	94%
17	2.857142857	1	240	0.37	64%
18	4.047619048	0	360	-0.12	45%
19	3.80952381	0	300	-0.04	48%
20	3.333333333	0	300	-0.31	38%
21	3.571428571	1	240	0.78	78%
22	4.285714286	1	300		0.93 82%
23	4.523809524	1	240		1.33 91%
24	3.571428571	1	180		1.13 87%
25	4.285714286	1	360		0.71 76%

Figure 4.11: SUM score calculation based on user answers for Task-2.

Summary

	SUM	Completion	Sat	Time
High	94.4%	95.1%	92.5%	95.5%
Avg.	88.5%	84.3%	85.0%	90.0%
Low	78.5%	73.5%	77.4%	84.5%

Task Raw Data

User	Satisfaction	Completion	Times	Standardized A	% Avg
1	3.095238095	1	120	0.47	68%
2	4.047619048	1	300	0.19	58%
3	5	1	240	1.03	85%
4	5	1	240	1.03	85%
5	5	1	420	0.48	69%
6	5	1	300	0.81	79%
7	4.761904762	1	240	0.88	81%
8	4.285714286	1	180	0.85	80%
9	5	1	240	1.03	85%
10	5	1	180	1.31	91%
11	4.047619048	0	240	-0.50	31%
12	5	0	240	0.12	55%
13	4.285714286	1	120	1.25	89%
14	4.523809524	1	180	1.01	84%
15	4.761904762	1	180	1.16	88%
16	4.761904762	1	180	1.16	88%
17	4.523809524	1	240	0.72	77%
18	5	1	120	1.71	96%
19	4.285714286	1	120	1.25	89%
20	4.285714286	1	180	0.85	80%
21	3.571428571	1	180	0.39	65%
22	5	1	180		
23	4.047619048	1	120	1.31	91%
24	5	1	240	1.09	86%
25	4.761904762	0	120	1.03	85%
				0.64	74%

Figure 4.12: SUM score calculation based on user answers for Task-3.

Summary

	SUM	Completion	Sat	Time
High	96.1%	97.3%	94.7%	96.4%
Avg.	91.5%	87.9%	88.5%	91.6%
Low	82.5%	78.5%	82.3%	86.7%

Task Raw Data

User	Satisfaction	Completion	Times	Standardized A	% Avg
1	3.333333333	1	180	0.42	66%
2	4.285714286	1	480	0.33	63%
3	4.761904762	1	420	0.79	78%
4	4.761904762	1	300	1.05	85%
5	5	1	540	0.76	78%
6	4.761904762	1	300	1.05	85%
7	4.761904762	1	480	0.68	75%
8	4.285714286	1	240	0.89	81%
9	5	1	360	1.08	86%
10	5	1	300	1.23	89%
11	4.047619048	0	360	-0.63	26%
12	4.523809524	0	360	-0.29	39%
13	4.523809524	1	180	1.29	90%
14	5	1	180	1.64	95%
15	5	1	240	1.41	92%
16	4.761904762	1	240	1.23	89%
17	4.523809524	1	300	0.88	81%
18	5	1	120	1.96	97%
19	4.523809524	1	120	1.61	95%
20	4.285714286	1	240	0.89	81%
21	3.571428571	1	240	0.37	64%
22	5	1	240		
23	4.047619048	1	240	1.41	92%
24	5	1	300	0.71	76%
25	4.761904762	1	120	1.23	89%
				1.79	96%

Figure 4.13: SUM score calculation based on user answers for Task-4.

Copyright © 2011 Measuring Usability LLC <http://www.measuringusability.com>

Summary				
	SUM	Completion	Sat	Time
High	81.1%	60.1%	84.7%	98.3%
Avg.	71.9%	44.6%	74.0%	95.5%
Low	61.7%	29.1%	63.3%	92.6%

Task Raw Data					
User	Satisfaction	Completion	Times	Standardized A	% Avg
1	4.761904762	0	180	0.60	73%
2	4.523809524	0	240	0.13	55%
3	4.523809524	0	180	0.49	69%
4	4.523809524	0	180	0.49	69%
5	5	1	300	0.74	77%
6	5	1	180	1.39	92%
7	4.523809524	1	180	1.16	88%
8	3.80952381	1	180	0.82	79%
9	4.761904762	1	120	1.78	96%
10	4.761904762	0	120	1.11	87%
11	4.047619048	0	180	0.26	60%
12	5	0	180	0.72	76%
13	5	1	120	1.90	97%
14	5	1	180	1.39	92%
15	5	1	120	1.90	97%
16	3.571428571	0	180	0.03	51%
17	5	1	120	1.90	97%
18	5	0	120	1.23	89%
19	3.571428571	0	180	0.03	51%
20	3.80952381	1	240	0.46	68%
21	4.761904762	0	240	0.24	60%
22	2.142857143	0	120	-0.14	44%
23	3.80952381	0	180	0.15	56%
24	5	1	240	1.03	85%
25	5	0	180	0.72	76%

Figure 4.14: SUM score calculation based on user answers for Task-1.

Copyright © 2011 Measuring Usability LLC <http://www.measuringusability.com>

Summary				
	SUM	Completion	Sat	Time
High	79.3%	63.8%	78.3%	95.9%
Avg.	68.9%	48.2%	66.1%	90.6%
Low	57.3%	32.6%	53.9%	85.3%

Task Raw Data					
User	Satisfaction	Completion	Times	Standardized A	% Avg
1	5	0	240	0.70	76%
2	4.523809524	1	360	0.61	73%
3	4.523809524	0	300	0.18	57%
4	3.80952381	0	180	0.48	69%
5	5	1	360	0.84	80%
6	5	1	240	1.37	91%
7	4.523809524	1	300	0.84	80%
8	3.80952381	1	300	0.49	69%
9	4.761904762	1	300	0.96	83%
10	4.761904762	0	240	0.58	72%
11	4.047619048	0	240	0.23	59%
12	5	0	300	0.41	66%
13	4.285714286	1	120	1.91	97%
14	5	1	240	1.37	91%
15	4.761904762	1	180	1.62	95%
16	3.571428571	0	300	-0.30	38%
17	5	1	240	1.37	91%
18	4.285714286	0	180	0.72	76%
19	3.571428571	0	240	-0.01	50%
20	3.80952381	1	300	0.49	69%
21	3.80952381	0	240	0.11	54%
22	2.142857143	0	180	-0.34	37%
23	3.80952381	0	300	-0.18	43%
24	4.523809524	1	300	0.84	80%
25	4.047619048	0	300	-0.06	48%

Figure 4.15: SUM score calculation based on user answers for Task-2.

Copyright © 2011 Measuring Usability LLC. <http://www.measuringusability.com>

Summary				
	SUM	Completion	Sat	Time
High	91.2%	89.9%	88.2%	95.5%
Avg.	83.7%	77.1%	78.6%	89.9%
Low	72.5%	64.2%	69.1%	84.3%

Task Raw Data					
User	Satisfaction	Completion	Times	Standardized A	% Avg
1	3.80952381	0	540	-0.91	18%
2	5	1	240	1.28	90%
3	4.047619048	0	300	-0.28	39%
4	5	1	240	1.28	90%
5	5	1	420	0.82	79%
6	4.523809524	1	300	0.80	79%
7	4.523809524	1	360	0.65	74%
8	4.761904762	1	360	0.80	79%
9	4.761904762	1	180	1.37	92%
10	5	0	300	0.30	62%
11	5	1	180	1.52	94%
12	4.761904762	0	180	0.58	72%
13	5	1	420	0.82	79%
14	4.761904762	1	180	1.37	92%
15	4.761904762	1	180	1.37	92%
16	4.047619048	1	240	0.70	76%
17	3.095238095	1	180	0.35	64%
18	4.523809524	0	180	0.43	67%
19	3.80952381	1	120	1.12	87%
20	4.047619048	1	240	0.70	76%
21	4.285714286	1	360	0.51	69%
22	3.333333333	1	240	0.26	60%
23	5	1	240	1.28	90%
24	4.285714286	1	120	1.42	92%
25	4.285714286	1	480	0.27	61%

Figure 4.16: SUM score calculation based on user answers for Task-3.

Copyright © 2011 Measuring Usability LLC. <http://www.measuringusability.com>

Summary				
	SUM	Completion	Sat	Time
High	93.5%	95.1%	87.5%	98.0%
Avg.	87.6%	84.3%	77.7%	94.7%
Low	77.6%	73.5%	67.9%	91.5%

Task Raw Data					
User	Satisfaction	Completion	Times	Standardized A	% Avg
1	3.571428571	0	660	-1.00	16%
2	4.523809524	1	360	1.14	87%
3	4.047619048	1	420	0.66	74%
4	4.761904762	1	480	1.06	85%
5	4.761904762	1	480	1.06	85%
6	4.523809524	1	480	0.88	81%
7	4.523809524	1	720	0.52	70%
8	4.523809524	1	480	0.88	81%
9	4.761904762	1	300	1.47	93%
10	4.761904762	1	480	1.06	85%
11	4.761904762	1	300	1.47	93%
12	4.761904762	1	300	1.47	93%
13	4.285714286	1	600	0.51	70%
14	4.761904762	1	240	1.67	95%
15	4.761904762	1	300	1.47	93%
16	4.285714286	1	300	1.13	87%
17	3.333333333	1	300	0.44	67%
18	4.523809524	0	180	0.84	80%
19	3.80952381	0	240	0.07	53%
20	4.047619048	1	480	0.54	71%
21	4.285714286	1	480	0.71	76%
22	3.333333333	1	300	0.44	67%
23	5	1	300	1.65	95%
24	4.285714286	1	240	1.33	91%
25	4.285714286	1	720	0.35	64%

Figure 4.17: SUM score calculation based on user answers for Task-4.

As a result, a graph as seen in **Figure 4.18** was obtained. These test results are results in which users see only a single system. This change in user perception can be seen in the second graph of **Figure 4.19**.

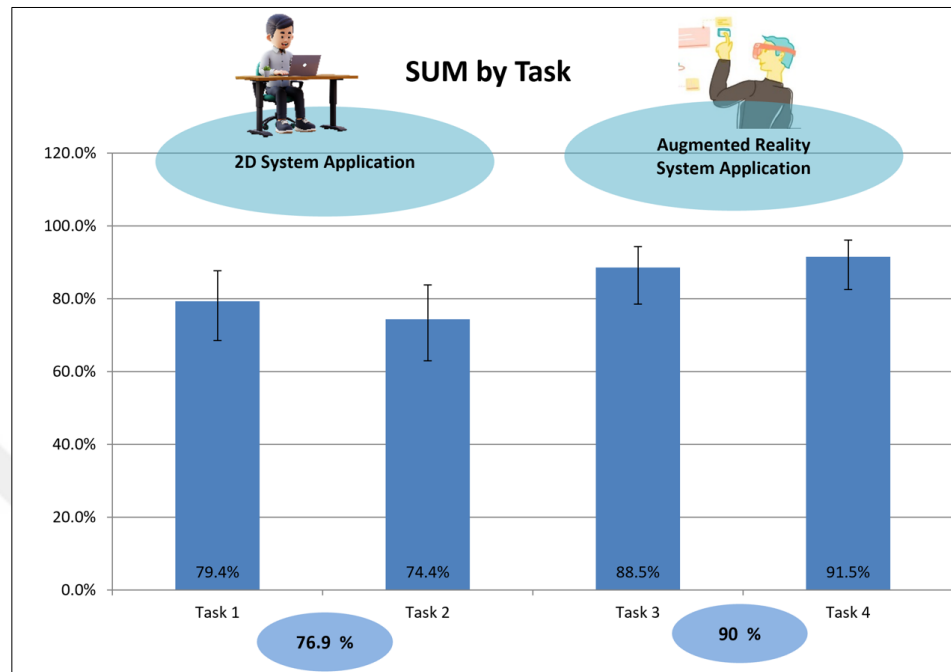


Figure 4.18: Comparison of system 1 and 2.

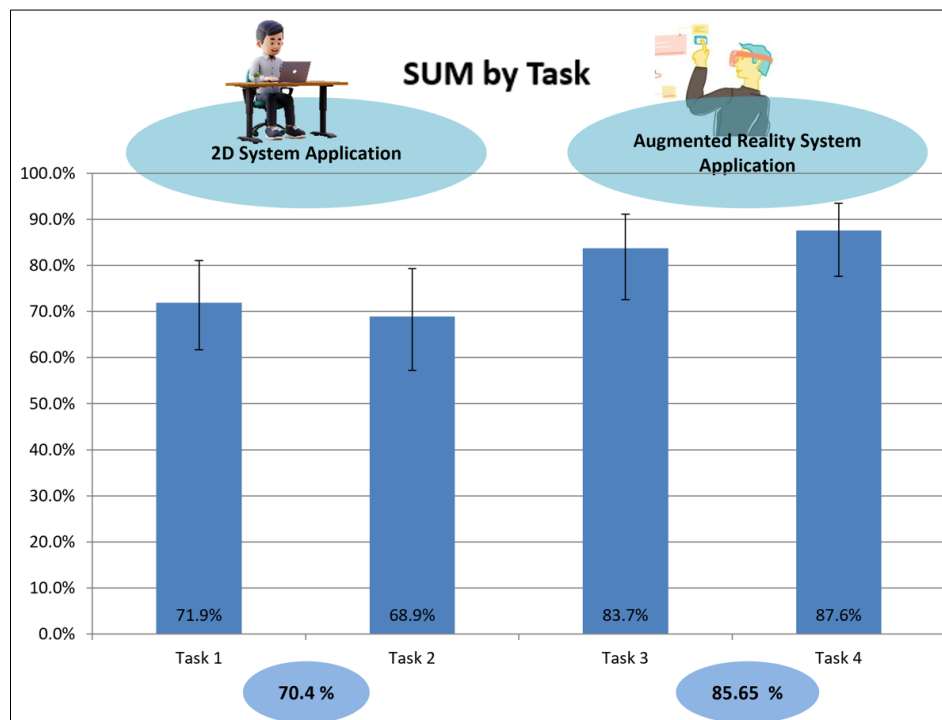


Figure 4.19: Comparison of system 1 and 2.

The result obtained from the first system seen; when the average of Task-1 and Task-2 is taken, it is found to be 76.9%. This is how the usability of 2D interfaces is measured. When the average of Task-3 and Task-4 is taken, it is found as 90%. The result obtained from the use of augmented reality interfaces in a 3D environment was measured in this way. The impact of switching from a 2D system to an augmented reality system and vice versa on usability was also examined. Firstly, in the analysis conducted among users who switched from the 2D system to the AR system, the usability rate increased from 76.9% to 85.65%. This shows that augmented reality technology increases usability for this user group. Looking at users' comments on this subject; It has been mentioned that augmented reality is easier to understand, is more interactive, and enables the visualization of visual attributes.

In another scenario, in the transition from the augmented reality system to the 2D system, the usability rate decreased from 90% to 70.4%. In this case, it is seen that AR systems provide a higher usability than 2D systems for some users, but augmented reality technology is not preferred by everyone or does not provide the same level of usability. Looking at the comments of the users on this issue, the ease of accessing the 2D interfaces and the difficulties in clicking the virtual object on the Hololens device were mentioned.

Users were very satisfied with the experience of interacting with virtual objects. The general comment received from users interested in deep learning was that they needed such a system to examine the models they created. When looking at the answers to other questions asked to users; Questions and answers about 2D Interfaces are shown in **Figure 4.20**. Questions and answers about AR are shown in **Figure 4.21**.

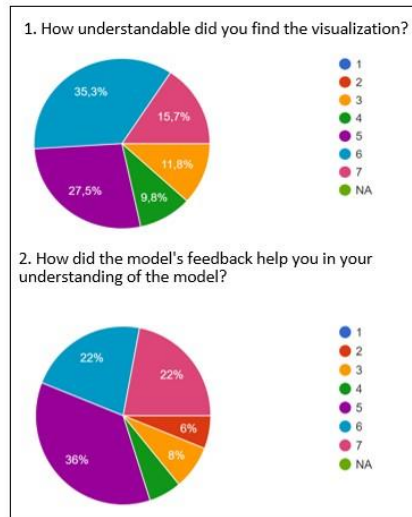


Figure 4.20: User answers to 2D interface questions.

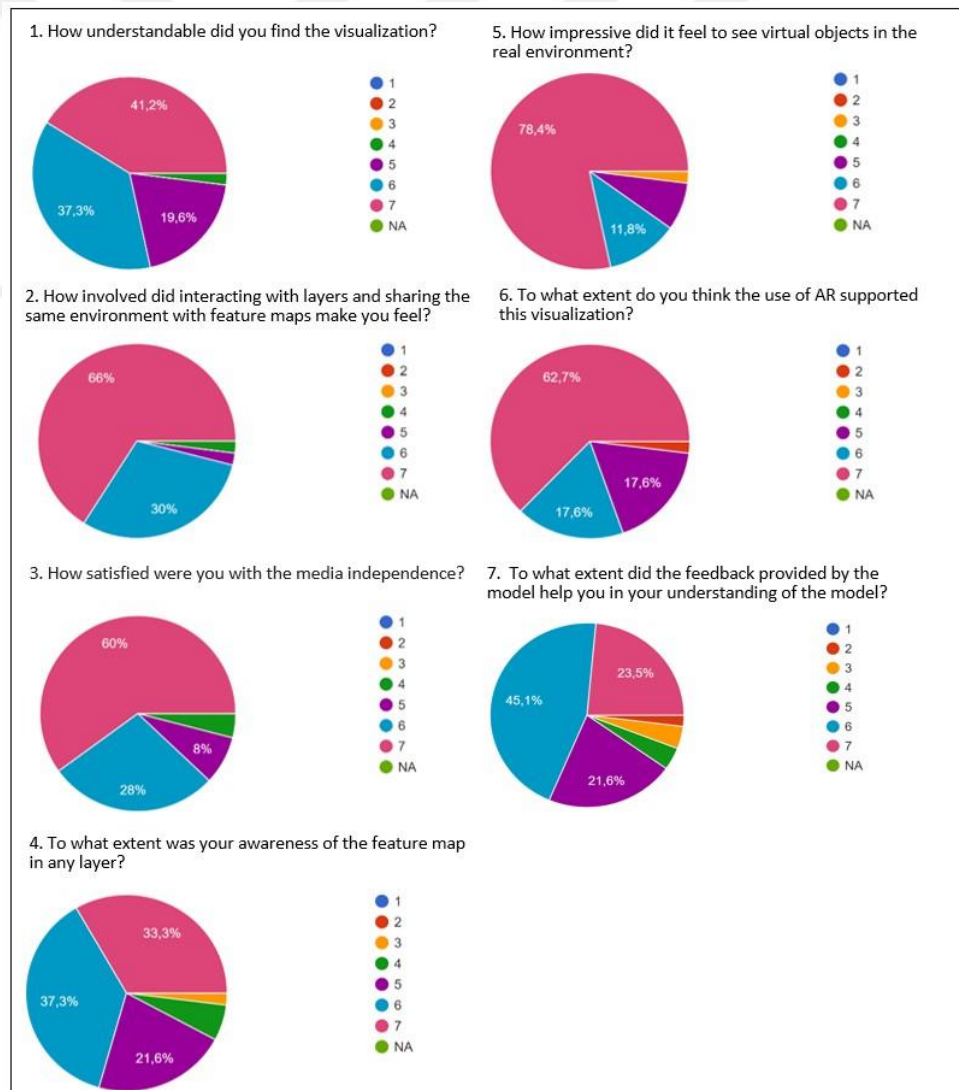


Figure 4.21: User answers to augmented reality questions.

5. CONCLUSION

In the study, the effect of using augmented reality interfaces in deep learning models on the user's perception was investigated and the current system was taken as a starting point. The fact that the subject in question is the user has led to a focus on user experience research. At this stage, what is being done has been examined and a road map has been drawn. The needs of the users were determined and an application was made accordingly. The version of the same application that runs on computers, which is the system that users have used so far, and the version that runs on the Hololens-2 device, which allows viewing of augmented reality interfaces, has been prepared.

Control and experimental groups were created with A/B testing and each group was given a single system experience. Surveys and measurements were made during and at the end of the test. Usability measurements were made using the SUM model, and satisfaction with this measurement was evaluated with ASQ questions. According to the results, the usability of augmented reality interfaces was measured as 90%, while this rate was 76.9% for interfaces in 2D environments. After being shown a single system, the users who were given the experience of the other system were asked a single question at the end of the survey about which system they would prefer, and 83.3% of the answers were received as an augmented reality system. In the analysis conducted with those who used the augmented reality system after the 2-dimensional system, the usability, which was 76.9% in 2 dimensions, increased to 85.65% with the same users. In the other case, the usability, which was 90% in the augmented reality system, was measured as 70.4% in 2 dimensions. Additionally, different questions asked to users revealed how impressed they were by this experience. The comfort provided by the limitless environment and the awareness created by interacting with the objects have increased the perception ability of the users. This shows that the emerging technology can provide a significant benefit in deep learning models.

This study is about the visualization of deep learning models, which we consider as artificial intelligence systems. This visualization was made with augmented reality interfaces. The study focused on the understandability of deep learning models. The aim of the study is to investigate user experience with augmented reality interfaces in

user perception while visualizing deep learning models and to explore innovative research areas in AR and DL. This study appears to be the first study conducted in this field.

Based on the data obtained from the research, it was concluded that this technology can enable people to be more effective or efficient in complex deep learning models and people to develop their existing talents by supporting them. In the later stages of the study;

- In addition to the understandability of the model, we can also look at its explainability.
- Work can be done to explain more complex deep learning models.
- Testing processes can be expanded by adding tasks for this purpose.

REFERENCES

- [1] Guzman M. d., (2002), “The role of visualization in the teaching and learning of mathematical analysis”.
- [2] DOĞAN F., TÜRKOĞLU İ., (2019), “Derin öğrenme modelleri ve uygulama alanlarına ilişkin bir derleme”, Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi, 10(2), 409–445.
- [3] Fuhrmann A., Loffelmann H., Schmalstieg D., Gervautz M., (1998), “Collaborative visualization in augmented reality”, IEEE Computer Graphics and Applications, 18(4), 54–59.
- [4] BAKAN, İsmail KEFE, İlker (2012), “Kurumsal açıdan algı ve algı yönetimi”, Kahramanmaraş Sütçü İmam Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi, 2(1), 19–34.
- [5] Reardon, Ann Partridge 2004. A Perceptual Basis for the Elements and Principles of Design, Eastern Michigan University, PhD (n.d.).
- [6] ÖZKİRİŞÇİ I. H., (2020), “Algı ve zaman bağlamında grafik imge”, Sanat ve Tasarım Dergisi,(25), 251–273.
- [7] Koffka, K. (1935), “Principles of Gestalt psychology Harcourt”, Brace & World. Inc. New.
- [8] Olujimi A., Priscilla E., Amanze R., Kuyoro S., A.O. A., (2022), “User-centered/user experience uc/ux design thinking approach for designing a university information management system”, Ingénierie des systèmes d information, 27, 577–590.
- [9] Wolcott M. D., McLaughlin J. E., (2023), “Exploring user experience (ux) Research methods in health professions education”, Currents in Pharmacy Teaching and Learning.
- [10] Şahin B., Akarçay E., Ateş O., Yıldırım F., Dürmüş B., (2024), “Tasarım odaklı düşünme yönteminden yararlanarak artırılmış gerçeklik ile kullanıcı deneyiminin geliştirilmesi”.
- [11] Yulia R., Candra R. M., Irsyad M., Darmizal T., (2023), “Ui/ux redesign of inhildukcapil application using the design thinking method”, INFOKUM, 10(5), 481–488.
- [12] Barnum C., (2019), “The state of ux research”, Journal of Usability Studies, 15(1), 1–7.
- [13] Marcus, A. (Ed.). (2014). Design, User Experience, and Usability: User Experience Design for Diverse Interaction Platforms and Environments: Third International Conference, DUXU 2014, Held as Part of HCI International 2014,

Heraklion, Crete, Greece, June 22-27, 2014, Proceedings, Part II (Vol. 8518).

[14] Gibbons, S. (2016), “Design Thinking 101”, <https://www.nngroup.com/articles/design-thinking/>, (2024).

[15] Nielsen J., (2004), “Usability engineering”, Computer Science Handbook, Second Edition, 45–1.

[16] Sauro J., Lewis J. R., (2012), “Quantifying the User Experience: Practical Statistics for User Research”, 1st Edition, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA

[17] McGee M., (2003), “Usability magnitude estimation”, Proceedings of the Human Factors and Ergonomics Society Annual Meeting, 47(4), 691–695.

[18] Kirakowski J., Corbett M., (2006), “Sumi: the software usability measurement inventory”, British Journal of Educational Technology, 24, 210 – 212.

[19] Chin J., Diehl V., Norman K., (1988), “Development of an instrument measuring user satisfaction of the human-computer interface”, In Proceedings of the SIGCHI conference on Human factors in computing systems (pp. 213-218).

[20] Lewis J., (1992), “Psychometric evaluation of the post-study system usability questionnaire: The pssuq”, In Proceedings of the human factors society annual meeting (Vol. 36, No. 16, pp. 1259-1260). Sage CA: Los Angeles, CA: Sage Publications.

[21] Lewis J., (1991), “Psychometric evaluation of an after-scenario questionnaire for computer usability studies: The asq”, SIGCHI Bull., 23, 78–81.

[22] Sauro, J., & Kindlund, E. (2005, July). “Using a single usability metric (SUM) to compare the usability of competing products”, In Human Computer Interaction International Conference.

[23] Sauro J., Kindlund E., (2005), A method to standardize usability metrics into a single score, In Proceedings of the SIGCHI conference on Human factors in computing systems (pp. 401-409).

[24] Cipresso P., Chicchi Giglioli I., Alcañiz Raya M., Riva G., (2018), “The past, present, and future of virtual and augmented reality research: A network and cluster analysis of the literature”, Frontiers in Psychology, 9, 2086.

[25] Azuma R., Baillot Y., Behringer R., Feiner S., Julier S., Macintyre B., (2001), “Recent advances in augmented reality. ieee comput graphics appl”, Computer Graphics and Applications, IEEE, 21, 34 – 47.

[26] Carmigniani J., Furht B., Anisetti M., Ceravolo P., Damiani E., Ivkovic M., (2010), “Augmented reality technologies, systems and applications”, Multimedia Tools and Applications, 51, 341–377.

- [27] Zhou F., Duh H., Billingham M., (2008), "Trends in augmented reality tracking, interaction and display: A review of ten years of ismar", 2008 7th IEEE/ACM International Symposium on Mixed and Augmented Reality, 2, 193–202.
- [28] Ababsa F.-E., Mallem M., (2008), "Robust camera pose estimation combining 2d/3d points and lines tracking", In 2008 IEEE International Symposium on Industrial Electronics (pp. 774-779).
- [29] Malik S., McDonald C., Roth G., (2002), "Hand tracking for interactive pattern-based augmented reality", In Proceedings. International Symposium on Mixed and Augmented Reality (pp. 117-126). IEEE.
- [30] Renner P., Pfeiffer T., (2017), "Attention guiding techniques using peripheral vision and eye tracking for feedback in augmented-reality-based assistance systems", In 2017 IEEE symposium on 3D user interfaces (3DUI) (pp. 186-194).
- [31] Ishii H., Ullmer B., (1998), "Tangible bits: Towards seamless interfaces between people, bits and atoms", In Proceedings of the ACM SIGCHI Conference on Human factors in computing systems (pp. 234-241).
- [32] Azuma R., Baillot Y., Behringer R., Feiner S., Julier S., MacIntyre B., (2001), "Recent advances in augmented reality", IEEE Computer Graphics and Applications, 21(6), 34–47.
- [33] Deng L., Yu D., (2014), "Deep learning: Methods and applications", Foundations and trends® in signal processing, 7(3–4), 197-387.
- [34] LeCun Y., Bengio Y., Hinton G., (2015), "Deep learning", Nature, 521(7553), 436–44
- [35] Schulz H., Behnke S., (2012), "Deep learning", KI - Künstliche Intelligenz, 26, 357-363.
- [36] Schmidhuber J., (2014), "Deep learning in neural networks: An overview", Neural Networks, 61, 85-117.
- [37] Trestioreanu L., Glauner P., Meira J., Gindt M., State R., (2020), "Using Augmented Reality and Machine Learning in Radiology", Technologies for Market Leadership: Investing in the Future, 89-106.
- [38] Lin P.-H., Chen S.-Y., (2020), "Design and evaluation of a deep learning recommendation based augmented reality system for teaching programming and computational thinking", IEEE Access, 8, 45689-45699.
- [39] Bermejo, C., Huang, Z., Braud, T., & Hui, P. (2017, June), "When augmented reality meets big data", In 2017 IEEE 37th International Conference on Distributed Computing Systems Workshops (ICDCSW) (pp. 169-174).
- [40] Yu R., Shi L., (2018), "A user-based taxonomy for deep learning visualization", Visual Informatics, 2(3), 147-154.

- [41] Neptune AI, “<https://neptune.ai/>” (2024).
- [42] Wandb, “<https://wandb.ai/wandb>” (2024).
- [43] TensorBoard, “<https://www.tensorflow.org/tensorboard>” (2024).
- [44] Visdom, “<https://ai.meta.com/tools/visdom/>”, (2024).
- [45] Hiplot, “<https://facebookresearch.github.io/hiplot/>”, (2024).
- [46] CNNVis, “<http://shixialiu.com/publications/cnnvis/demo/>”, (2024).
- [47] Neural Network Playground, “<https://playground.tensorflow.org/>”, (2024).
- [48] Netron, “<https://github.com/lutzroeder/netron>”, (2024).
- [49] ANN Visualizer, “<https://github.com/RedaOps/ann-visualizer>” (2024).
- [50] Erhan D., Bengio Y., Courville A., Vincent P., (2009), “Visualizing higher-layer features of a deep network”, Univeristé de Montréal, 1341(3), 1.
- [51] Simonyan K., Vedaldi A., Zisserman A., (2013), “Deep inside convolutional networks: Visualising image classification models and saliency maps”, arXiv preprint arXiv:1312.6034.
- [52] Springenberg J., Dosovitskiy A., Brox T., Riedmiller M., (2014), “Striving for simplicity: The all convolutional net”, arXiv preprint arXiv:1412.6806.
- [53] Smilkov D., Thorat N., Kim B., Viégas F., Wattenberg M., (2017), “Smoothgrad: removing noise by adding noise”, arXiv preprint arXiv:1706.03825.
- [54] Inkarebekov M., Monahan R., Pearlmutter B., (2023), “Visualization of ai systems in virtual reality: A comprehensive review”, arXiv preprint arXiv:2306.15545.
- [55] Bock M., Schreiber A., (2018), Visualization of neural networks in virtual reality using unreal engine, In Proceedings of the 24th ACM symposium on virtual reality software and technology (pp. 1-2).
- [56] Yosinski J., Clune J., Nguyen A., Fuchs T., Lipson H., (2015), “Understanding neural networks through deep visualization”, arXiv preprint arXiv:1506.06579.
- [57] Kath H., Lüers B., Gouvêa T., Sonntag D., (2023), “A virtual reality tool for representing, visualizing and updating deep learning models”, arXiv preprint arXiv:2305.15353.
- [58] Linse C., Alshazly H. A., Martinetz T., (2022), “A walk in the black-box: 3d visualization of large neural networks in virtual reality”, Neural Computing and Applications, 34(23), 21237–21252.

- [59] Aamir A., Tamosiunaite M., Wörgötter F., (2022), “Caffe2unity: Immersive visualization and interpretation of deep neural networks”, *Electronics*, 11(1), 83.
- [60] VanHorn K., Zinn M., Cobanoglu M., (2019), “Deep learning development environment in virtual reality”, *arXiv preprint arXiv:1906.05925*.
- [61] Lecun Y., Bottou L., Bengio Y., Haffner P., (1998), “Gradient-based learning applied to document recognition”, *Proceedings of the IEEE*, 86(11), 2278 – 2324.
- [62] Balasubramaniam S., Velmurugan Y., Jaganathan D., Dhanasekaran S., (2023), “A modified lenet cnn for breast cancer diagnosis in ultrasound images”, *Diagnostics*, 13(17), 2746.
- [63] Krizhevsky A., Sutskever I., Hinton G., (2012), “Imagenet classification with deep convolutional neural networks”, *Advances in neural information processing systems*, 25.
- [64] Guo H.-J., Prabhakaran B., (2022), “Hololens 2 technical evaluation as mixed reality guide”, *arXiv preprint arXiv:2207.09554*.
- [65] DeepLearningModelVisualization, “<https://github.com/busraylmz/DeepLearningModelVisualization.git>” (2024).
- [66] Figma Tool, “<https://www.figma.com/wireframe-tool/>”, (2024).

BIOGRAPHY

Büşra ÖZTÜRK graduated from Gazi University, Faculty of Engineering, Department of Computer Engineering, which she started in 2012, in 2017. In 2020, she started her master's degree at Gebze Technical University Graduate Education Institute, Department of Computer Engineering. She works as a software engineer at Havelsan Inc. Naval Combat Management Technologies Center.



PUBLICATIONS AND PRESENTATIONS FROM THE THESIS

Öztürk B., Genç Y., (2023), “Using augmented reality interfaces for artificial intelligence systems”, 12th International Conference on Soft Computing, Artificial Intelligence and Applications, 207–222, Sydney, Australia, 23-24 December.

