Interdisciplinary Postgraduate Program - Master of Science in Intelligent Systems

INTELLIGENT INFORMATION SYSTEMS

Master thesis with the title

Gesture Recognition using Artificial Intelligence and Application to an Unmanned Ground Vehicle (UGV)

Student: Christina Tsiftsi

Professor in charge: Nikolaos Papadakis

ATHENS 2024

This page was intentionally left blank.

Η Μεταπτυχιακή Διατριβή της **Χριστίνας Τσιφτσή** εγκρίνεται:

ΤΡΙΜΕΛΗΣ ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ

Καθηγητής **ΝΙΚΟΛΑΟΣ ΠΑΠΑΔΑΚΗΣ** (Επιβλέπων) Νικόλαος Παπαδάκης

Καθηγητής **ΝΙΚΟΛΑΟΣ ΔΑΡΑΣ**

Καθηγητής **ΝΙΚΟΛΑΟΣ ΜΑΤΣΑΤΣΙΝΗΣ** Nikolaos Matsatsinis Digitally signed by Nikolaos Matsatsinis Date: 2024.07.11 17:38:33 +03'00'

ΑΘΗΝΑ 2024

This page was intentionally left blank.

ACKNOWLEDGMENTS

I would like to express my deepest gratitude and appreciation to my supervisor, Nikolaos Papadakis, whose dedication, expertise, and unwavering support have been indispensable throughout the journey of completing this master thesis. His insightful guidance, constructive feedback, and encouragement have not only shaped the trajectory of this research but also contributed significantly to my academic and personal growth. Furthermore, I would like to express my heartfelt gratitude to my family and friends for their unwavering support, understanding, and encouragement during the challenging moments of this academic endeavor. Their belief in my abilities has been a constant source of motivation and inspiration. Lastly, I extend my gratitude to the participants of this study whose willingness to share their insights and experiences has been integral to the success of this research endeavor.

This thesis is dedicated to my father. Though he is no longer with us, his constant care and dedication have been pivotal in bringing me to where I am today.

This page was intentionally left blank.

ABSTACT

This thesis explores the integration of advanced gesture recognition technologies into the control systems of unmanned ground vehicles (UGVs), aiming to enhance their operability and user interaction. The research leverages a comprehensive approach, combining Artificial Intelligence (AI), Machine Learning (ML), and Deep Learning (DL) methodologies to develop a robust and efficient gesture recognition system. The study begins with the theoretical foundations of AI, ML, and DL, establishing a framework for the subsequent development of gesture recognition algorithms, then delves into a thorough review of existing UGV control interfaces and identifies the limitations of traditional input methods. The core of the research involves the creation of a tailored gesture recognition system utilizing ML and DL techniques. This system is trained on diverse datasets to ensure adaptability to various user gestures and environmental conditions. The integration of real-time processing ensures swift and accurate interpretation of gestures, facilitating seamless communication between the operator and the UGV.

Furthermore, the thesis investigates the practical implementation of the gesture recognition system on an unmanned ground vehicle prototype. Through a series of experiments and simulations, the effectiveness of the developed system is evaluated in terms of responsiveness, accuracy, and overall usability in diverse operational scenarios. The findings of this research contribute to the field of UGV control interfaces by providing a novel, AI-driven solution that significantly improves the human-UGV interaction paradigm. The study not only advances the theoretical understanding of gesture recognition within the context of unmanned systems but also offers practical insights into the integration of such technologies for real-world applications.

In conclusion, the thesis establishes the potential of combining AI, ML, and DL in the realm of gesture recognition for unmanned ground vehicles, paving the way for more intuitive and efficient control interfaces in the evolving landscape of autonomous systems.

KEY WORDS

Gesture recognition, computer vision, UGV, machine learning, deep learning, artificial intelligence.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF IMAGES

CHAPTER 1: INTRODUCTION

1.1 Brief history and evolution of AI

Artificial Intelligence (AI) encompasses the field of computer science dedicated to the intelligence of machines. In this context, an intelligent agent refers to a system that takes actions with the goal of maximizing its likelihood of success. AI involves the exploration of concepts that empower computers to perform tasks commonly associated with human intelligence. Fundamental principles in AI encompass reasoning, knowledge, planning, learning, communication, perception, and the capability to manipulate objects and move. This discipline constitutes the science and engineering behind the creation of intelligent machines, with a particular focus on intelligent computer programs.[1]

Simpler put, according to Leslie D., et. al. [2], “AI systems are algorithmic models that carry out cognitive or perceptual functions in the world that were previously reserved for thinking, judging, and reasoning human beings”.

AI research encompasses a range of areas, including search algorithms, knowledge graphs, natural language processing, expert systems, evolution algorithms, machine learning (ML), deep learning (DL), and more.

The evolution of AI involves three primary elements: perceptual intelligence, cognitive intelligence, and decision-making intelligence. Perceptual intelligence entails providing machines with basic human-like abilities such as vision, hearing, and touch. Cognitive intelligence, on the other hand, represents a more advanced skill set involving induction, reasoning, and knowledge acquisition, drawing inspiration from cognitive science and brain-like intelligence to imbue machines with thinking logic resembling that of humans [3].

In his renowned paper "Computing Machinery and Intelligence," Alan Turing raised the fundamental question: "Can machines think?" Turing acknowledged the challenge of defining thinking clearly, as it is a subjective behaviour. To address this, he proposed an indirect approach to determine if a machine can think—the Turing test. This test evaluates a machine's ability to exhibit intelligence at a level indistinguishable from human beings. If a machine passes this test, it is deemed qualified to be labelled as artificial intelligence (AI).

Figure 1 below briefly presents the history of Artificial Intelligence.

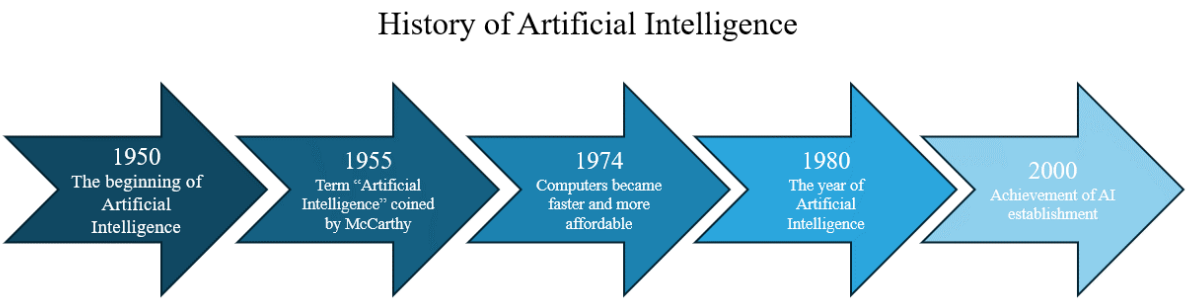


***Figure 1**: The history of AI.*

Around 1941, during the war, Alan Turing was contemplating machine intelligence. He distributed a typewritten paper, which unfortunately has been lost, about machine intelligence. This paper is regarded as "undoubtedly the earliest paper in the field of AI," according to Turing and Copeland [4]. Until 1950, the term "Artificial Intelligence" remained unfamiliar to many until John McCarthy, recognized as the pioneer of Artificial Intelligence, introduced "Artificial Intelligence" in 1955 through his published paper "Computing Machinery and Intelligence," which proposes the Turing Test as a way to determine whether a machine can demonstrate human-like intelligence. McCarthy, along with Alan Turing, Allen Newell, Herbert A. Simon, and Marvin Minsky, are hailed as one of the founding figures of AI, when in 1956 John McCarthy, Marvin Minsky, Nathaniel Rochester, and Claude Shannon organize the Dartmouth Conference, which is considered to be the birth of AI as a field of study. In 1965, Joseph Weizenbaum created ELIZA, a computer program capable of engaging in natural language conversations. This significant development marked an early milestone in the field of natural language processing. In 1974, the era of computers thrived. Over time, computers experienced a gradual surge, becoming faster, more cost-effective, and capable of storing larger amounts of information. Notably, they acquired the ability to think abstractly, self-recognize, and accomplish Natural Language Processing. 1980 marked the resurgence of AI research, fueled by increased funding and the development of advanced algorithmic tools. Deep learning techniques emerged, enabling computers to learn from user experiences, when in 1981, Terry Winograd developed SHRDLU, a pioneering program adept at comprehending and manipulating blocks within a virtual environment, showcasing an early example of a natural language understanding system. This breakthrough was followed by IBM's

Deep Blue's historic victory over the reigning world chess champion, Garry Kasparov, in 1997, marking the first instance of a computer defeating a world champion in a tournament setting. These milestones demonstrate significant advancements in artificial intelligence, from early natural language processing capabilities to the triumph of machine intelligence over human expertise in strategic games like chess. In the 2000s, AI finally reached significant milestones after numerous failed attempts. Despite limited government funding and public attention, AI flourished during this period.

1.2 Explanation of machine learning and its types

Machine Learning is a subset of Artificial Intelligence (AI) that involves the development of algorithms enabling machines to learn and enhance their performance without explicit programming for specific tasks[1]. In other words, a form of computing utilized for discerning patterns in data and making predictions for specific instances. The term "learning" can be deceptive, as the computer does not learn in the human sense. Rather, it identifies similarities and differences in data through repetitive parameter adjustments, often termed "training." As the information we give it changes, the computer also changes how it predicts things, learning to find new patterns. This process involves applying a mathematical formula to extensive input data, generating a corresponding outcome. In the realm of AI, Machine Learning (ML) exerts a significant and wide-ranging impact across various domains of technology and science[3]. Image 1 below presents a visual representation of Machine Learning.

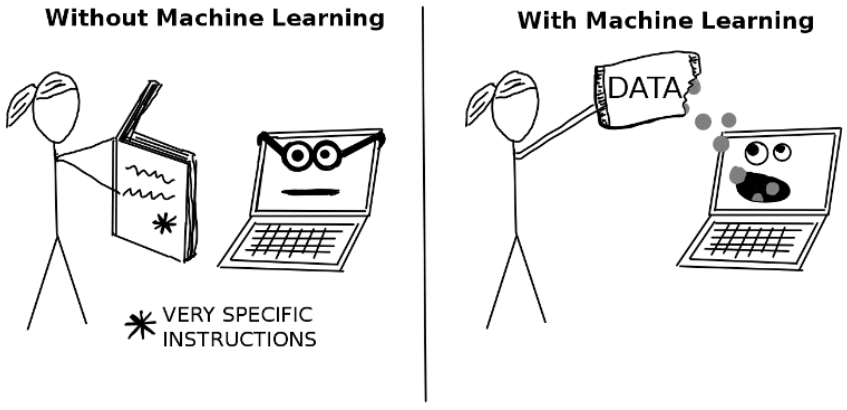


***Image 1**: Explanation of Machine Learning (Source: [37] https://christophm.github.io/interpretable-ml-book/terminology.html)*

Machine Learning basically can be divided into three types of algorithms: Supervised Learning (Task Driven), Unsupervised Learning (Data Driven) and Reinforcement Learning (Learning from Environment) as shown in figure 2 below:

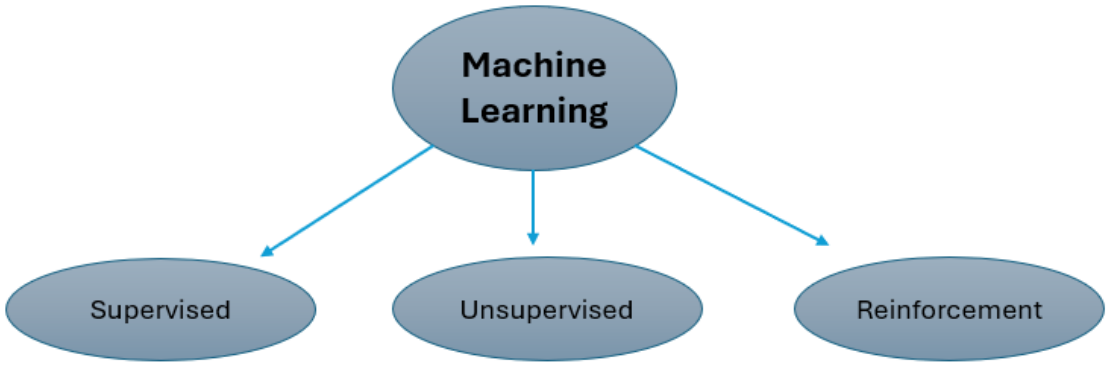


***Figure 2**: Types of ML algorithms.*

1.2.1 Supervised Learning

Supervised Learning is a method in machine learning where we teach a computer by showing it examples and telling it what those examples represent. We provide the computer with pairs of input (like images or data) and the correct output (like a label or a category). This helps the computer learn the relationship between inputs and outputs. For instance, if we're teaching the computer to recognize cats and dogs in pictures, we'd show it lots of pictures of cats and dogs along with labels saying which is which. The computer learns from these examples and can then predict the correct label for new pictures it hasn't seen before. If the outputs are categories (like "cat" or "dog"), it's called classification. If the outputs are numbers, it's called regression. It's different from Unsupervised Learning, where the computer learns without any labels, and Semi-supervised Learning, which uses a mix of labelled and unlabeled data. Additionally, in Active Learning, the computer might ask for help in learning by seeking feedback during the process [5].

In a simple machine learning model, the learning process consists of two main steps: training and testing. During the training process, the model learns from input data, which typically includes samples with features. These features are learned by the algorithm or learner, and they are used to construct the learning model. In other words, the model learns how to make predictions or

classifications based on the features it's given. Once the training is complete, the model is tested using test or production data. During testing, the model applies what it has learned to new data to make predictions. The output of this process is tagged data, which provides the final predictions or classifications generated by the model [6]. This process can be expressed in equation form as follows:

$$Y = f(X).$$

Where, Y is the output variable, X is the input variable, f () is the function that maps output to input.

Supervised Learning has a big advantage because it deals with meaningful classes or outputs that humans can understand easily, making it great for tasks like pattern classification and regression. However, it also has some downsides. One major issue is that it's hard to get labels for lots of input data, especially when there's a huge amount of it. For example, labeling a bunch of images for classification can be difficult. Plus, real-world concepts don't always have clear labels, which can make things uncertain. For instance, telling apart "hot" and "cold" isn't always straightforward, and naming something that's like both a loveseat and a bed can be tricky. These challenges can limit how useful Supervised Learning is in certain situations. To tackle these problems, we can look into other learning methods like Unsupervised Learning, Semi-supervised Learning, Reinforcement Learning, Active Learning, or even a mix of different approaches[5].

Algorithms used in supervised learning are Decision Trees, Linear Regression, Naive Bayes and Logistic Regression.

A. Decision Trees

A decision tree is like a flowchart that helps make decisions based on certain factors. It starts with a question at the top (the root) and branches out into more questions or outcomes (nodes) based on the answers. Each node tests a specific factor, splitting the data into smaller groups. Eventually, you reach the bottom of the tree (the leaves), which represent the final decisions or classifications. Figure 3 is an example of a decision tree constructed for understanding the risks to prevent a heart attack.

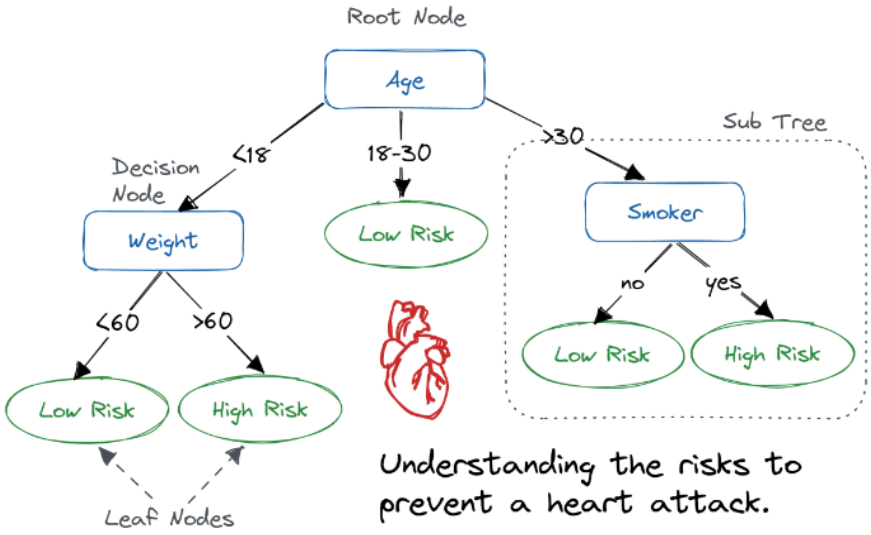


***Figure 3**: Example of Decision Tree (Datacamp.com)*

B. Linear Regression

Linear regression is a method used to understand the relationship between variables. It helps predict a continuous outcome (like price or temperature) based on input variables (like time or temperature). In simple terms, it's like drawing a straight line through data points to see how they're related. We use it in supervised learning, where we train the model on labeled data (data with known outcomes) and then use it to make predictions on new, unlabeled data [7].

In Figure 4, we see a model represented by a line. This model is calculated using training data, which are the points on the graph. Each point has a known label (shown on the y-axis). The goal of the model is to fit these points as closely as possible by minimizing the value of a chosen loss function. Once the model is trained, we can use it to predict unknown labels. In other words, if we only know the x-value, we can use the model to predict the corresponding y-value [8].

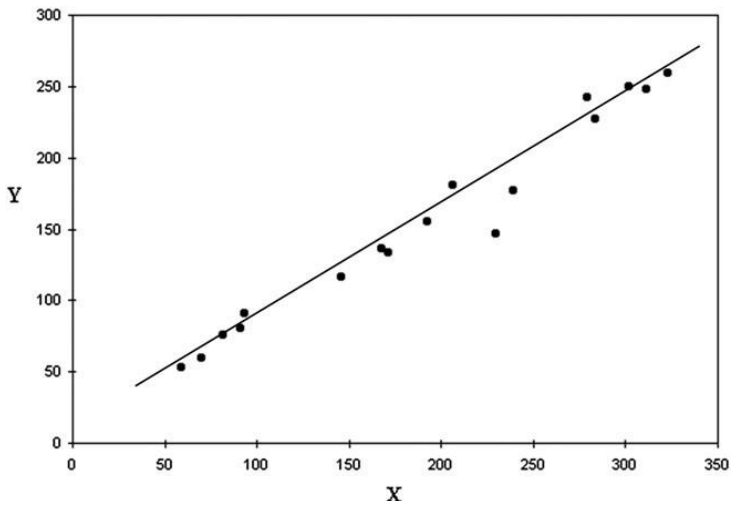


***Figure 4:** Linear Regression Representation*

C. *Naive Bayes*

Bayesian classification is a way of sorting things into categories using probabilities. It's like a methodical approach to handling uncertainty by assigning probabilities to different outcomes. This method helps solve prediction problems by combining observed data. It's practical and offers a useful perspective for understanding learning algorithms. Plus, it's good at dealing with noise in input data. According to Bayes, this conditional probability can be calculated [9] using the following formula:

$$P(A|B) = \frac{P(A|B)*P(A)}{\mathrm{P(B)}}$$

Where:

- P(B|A) = probability that event B occurs if event A has already occurred
- P(A) = probability that event A occurs
- P(B) = probability that event B occurs

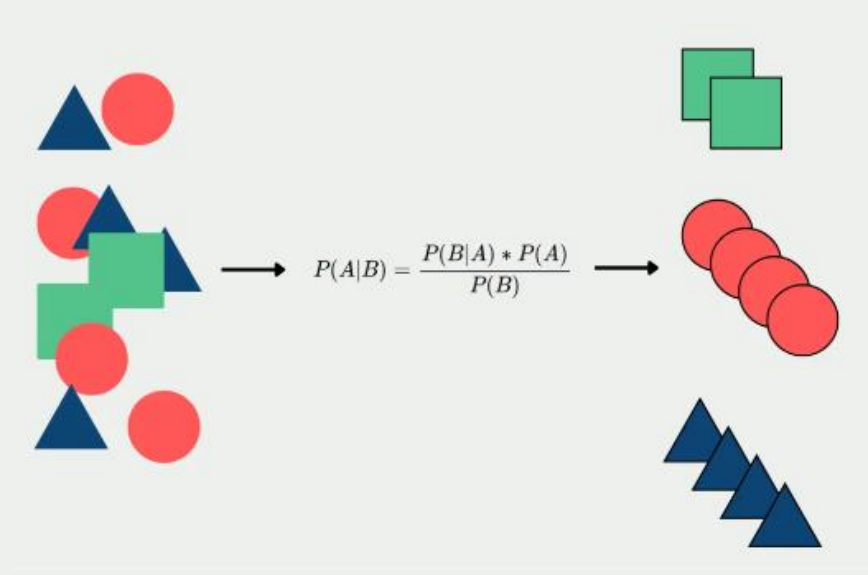


***Figure 5**: Simple Representation of the Naive Bayes Classification (Source databasecamp.de)*

D. *Logistic Regression*

Logistic regression, like naive Bayes, looks at the features in the input data and assigns weights to them. It then combines these features linearly by multiplying each one by a weight and adding

them up.[10] However, unlike naive Bayes, which is a generative classifier, logistic regression is a discriminative classifier. It predicts the probability of an event happening by fitting the data to a logistic function. It uses various predictor variables, whether numerical or categorical, to make these predictions.

Logistic regression indeed produces an S-shaped curve, also known as a sigmoid curve. The direction and steepness of this curve are influenced by changes in regression coefficients. A positive slope leads to an upward S-shaped curve, while a negative slope results in a downward S-shaped curve.

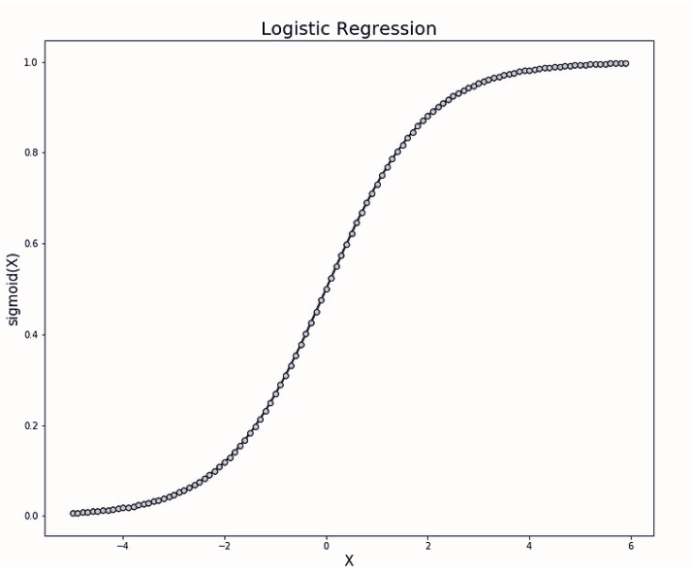


Figure 6: *Logistic Regression Representation (source: towardsdatascience.com)*

1.2.1.1 Applications of supervised learning

A. *Image and Object Recognition*: These models help locate, isolate, and categorize objects in videos or images, making them valuable for computer vision techniques and imagery analysis.

B. *Predictive Analytics:* Supervised learning is widely used in creating systems for predictive analytics. These systems offer deep insights into various business data points, enabling enterprises to anticipate outcomes based on specific variables, aiding decision-making.

C. Customer Sentiment Analysis: Organizations use supervised machine learning to extract and classify information, including context, emotion, and intent, from large data volumes with minimal human intervention. This enhances understanding of customer interactions and supports efforts to improve brand engagement.

D. *Spam Detection:* Supervised learning is applied to train databases to recognize patterns or anomalies in new data, effectively organizing spam and non-spam-related correspondences. [6]This helps in efficiently managing communication channels.

1.2.2 Unsupervised Learning

In unsupervised learning, the machine receives inputs but doesn't have target outputs or rewards. Despite this lack of feedback, it aims to create useful representations of the input data. These representations help in tasks like decision making or predicting future inputs. Unsupervised learning is about finding patterns in the data beyond just random noise. Two common examples are clustering, where similar data points are grouped together, and dimensionality reduction, which simplifies complex datasets [11]. According to Bengio et al. [12], unsupervised machine learning is considered as the avenue toward achieving genuine artificial intelligence. This form of AI comprehensively grasps the surrounding world and serves as a crucial element for the development of AI-generated designs and policies.

An advantage of unsupervised learning is that when it comes to predicting a binary outcome from a dataset, there are various well-established tools available, such as logistic regression, linear discriminant analysis, classification trees, and support vector machines. Plus, there are clear methods for assessing the quality of results, like cross-validation and validation on an independent test set.

However, unsupervised learning poses more challenges. It's often more subjective, and there isn't a straightforward goal like predicting a response. Additionally, assessing results from unsupervised learning methods can be tricky because there's no universally accepted way to perform cross-validation or validate results on an independent dataset. The main reason for this difference is simple: in supervised learning, we can check our work by seeing how well our model predicts the response on unseen data. But in unsupervised learning, there's no true answer to compare against since the problem is unsupervised [13].

A. *Clustering*

Clustering stands out as the most well-established subfield of unsupervised learning (UL). It discerns subgroups within unprocessed, unlabeled datasets by analyzing similarities and distinctions in their features [14]. Various clustering techniques are available, [15] with k-means

being one of the most notable. In this method, clustering is achieved by iteratively adjusting centroids and assigning data points to the nearest centroid, forming clusters where points share similarities within the same group. Differences between clusters are discernible, as each cluster represents a distinct set of properties. The number of clusters is predetermined by the user, allowing for flexibility in the analysis. Lloyd's [16] algorithm, also known as K-Means clustering, introduced a method for grouping data points based on their proximity using the Euclidean distance metric. This approach involves iteratively adjusting cluster centroids and assigning data points to the nearest centroid, effectively partitioning the data into clusters based on their similarity in Euclidean space.

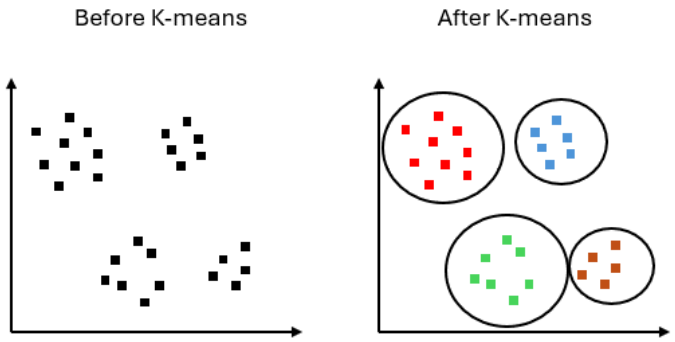


Figure 7: *K-means Clustering*

B. Anomaly detection

In data mining, anomaly detection, also known as outliers, noise, deviations, novelties, and exceptions, refers to identifying items, remarks, or events that do not conform to expected patterns [17]. Anomaly-based intrusion detection in networks involves identifying unusual patterns in network traffic that deviate from the expected normal behavior. These irregular patterns are often labeled as anomalies, outliers, exceptions, aberrations, surprises, peculiarities, or discordant observations across different application domains. Among these terms, "anomalies" and "outliers" are the most frequently used terms within the context of anomaly-based intrusion detection in networks [18].

C. Unsupervised neural networks

The concept of unsupervised learning in Artificial Neural Networks (ANNs) has the potential to revolutionize artificial systems. Unsupervised learning involves teaching a system to understand

input signals in a way that reveals their underlying patterns, without explicit guidance. Several studies have demonstrated the effectiveness of unsupervised learning algorithms across various types of neural networks.

Neural networks are computational models inspired by the nervous systems of animals, particularly the brain, known for their ability in pattern recognition and machine learning. They adapt their internal structure and data flow during training, making them highly versatile systems. Neural networks consist of three main layers: the input layer, which interacts with external data and sets the conditions for training; the hidden layer, positioned between the input and output layers, containing neurons that process information; and the output layer, which produces results or signals to the external environment. The number of neurons in the output layer corresponds to the tasks the neural network is designed to perform.

Neural networks [19] are capable of learning without the need for manual reprogramming, and they typically feature a layered architecture, as depicted in Figure 8.

Unsupervised learning refers to a network's ability to learn and represent input patterns in a manner that mirrors the overall arrangement of all input designs or patterns. It's a machine learning technique tasked with deducing hidden structures from unlabeled data. Essentially, this entails learning algorithms that lack labeled data to guide the training process. In unsupervised learning algorithms, a significant amount of data and characteristics of each observation are provided as inputs, but without corresponding desired outputs. Unsupervised learning is commonly utilized, for instance in clustering, to group images into sets or clusters based on inherent features such as color, size, and shape [3] [20].

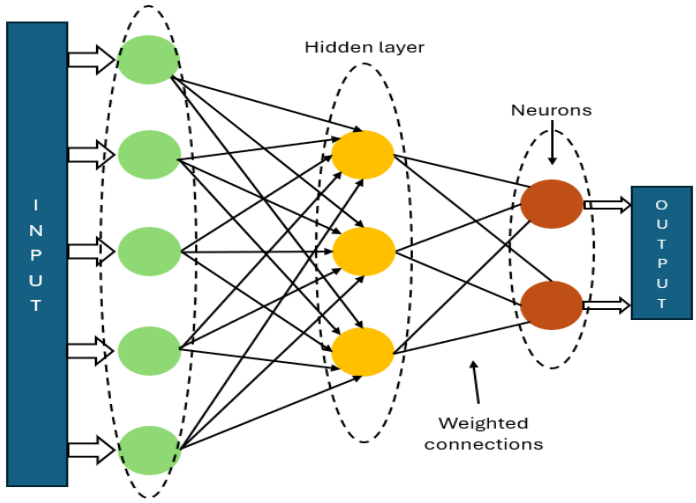


Figure 8: *The Layered architecture of ANN system (source: [19])*

1.2.2.1 Applications of unsupervised learning

A. Search Engines

Clustering is vital for search engines, organizing search results efficiently by assessing content similarity. K-means clustering, a key method, assumes documents with similar keywords and phrases better match search queries. [21] To evaluate similarity, search engines analyze each record's content, excluding less significant elements. The ranked content considers relevance, previous searches, and frequency of use, enhancing page ranking efficiency and improving the search experience.

B. Anomaly Detection for business analysis

In business analysis, K-means clustering serves as a valuable tool for anomaly detection, identifying outliers within a dataset. These outliers may represent abnormal time values, atypical user behavior, inconsistent experiment results, and more. For instance, in financial market analysis, K-means can detect repetitive transactions indicative of suspicious behavior, signaling potential fraud to banks or financial organizations [21].

C. Healthcare

Clustering methods present significant opportunities for mining data to unveil natural structures and patterns reflecting intricate human pathophysiology. To [22] enhance clinical care and research, effective clustering practices demand a comprehensive grasp of data processing and optimization, feature selection, evaluation of clustering methods' strengths and weaknesses, identification of the most suitable clustering approach, and its application to problem-solving in clinical contexts.

D. Social Network Analysis (SNA)

Social network analysis involves investigating qualitative and quantitative social structures through the application of Graph Theory, a significant branch of discrete mathematics, to networks. In this process, the social network structure is represented by nodes (individuals, personalities, or entities within the network) and edges or links (relationships, interactions, or communications) connecting them. Clustering methods play a crucial role in this analysis by facilitating the mapping and measurement of relationships and conflicts among people, groups,

companies, computer networks, and other connected entities, enabling a deeper understanding of social dynamics and network behavior [6].

1.2.3 Reinforcement Learning

In reinforcement learning, [11] a machine interacts with its environment by taking actions, which affect the state of the environment and result in the machine receiving rewards or punishments. The machine's goal is to learn how to act in a way that maximizes its future rewards or minimizes punishments over its lifetime.

1.2.3.1 Applications of reinforcement learning

A. Self-driving cars

Numerous research papers have advocated for the integration of Deep Reinforcement Learning (DRL) into the domain of autonomous driving. With self-driving cars, a multitude of considerations arise, ranging from adhering to speed limits and navigating drivable zones to avoiding collisions. Reinforcement learning offers promise in various autonomous driving tasks, encompassing trajectory optimization, motion planning, dynamic pathing, controller optimization, and scenario-based learning policies tailored for highway scenarios. For instance, DRL facilitates the learning of automatic parking policies and overtaking maneuvers while ensuring collision avoidance and maintaining consistent speeds, thereby demonstrating its potential in addressing the multifaceted challenges of autonomous driving.

B. Trading and finance

Supervised time series models have demonstrated their efficacy in predicting future sales and stock prices, but they fall short in prescribing actions at specific stock price thresholds. This deficiency is addressed by Reinforcement Learning (RL), enabling an RL agent to autonomously determine whether to hold, buy, or sell securities. The RL model is assessed against market benchmark standards to ensure its optimal performance. This automated approach instills consistency into the decision-making process, diverging from previous methods that heavily relied on manual analyst interventions for every decision. Notably, corporations like IBM [23] have implemented sophisticated RL-based platforms for financial trading, wherein the reward function is computed based on the profit or loss incurred in each financial transaction.

C. NLP (Natural Language Processing)

Within Natural Language Processing (NLP), Reinforcement Learning (RL) finds application in various tasks such as text summarization, question answering, and machine translation. In a paper authored by Eunsol Choi, Daniel Hewlett, and Jakob Uszkoreit, [24] they introduce an RL-based approach for question answering in lengthy texts. Their method involves initially selecting pertinent sentences from the document for answering the question, followed by utilizing a slow Recurrent Neural Network (RNN) to generate answers based on these selected sentences. Additionally, a blend of supervised and reinforcement learning techniques is employed for abstractive text summarization. Another paper, led by Romain Paulus, Caiming Xiong, and Richard Socher [25], addresses the challenge of summarization in longer documents using Attentional, RNN-based encoder-decoder models. Their proposed neural network incorporates a novel intra-attention mechanism that attends to the input while continuously generating output. Training methods in this paper involve a combination of standard supervised word prediction and reinforcement learning strategies.

1.3 Introduction to deep learning and neural networks

Deep learning is a crucial technology in the Fourth Industrial Revolution (Industry 4.0) and Web3. It involves creating and teaching neural networks and decision-making nodes. Unlike traditional methods where features in data are manually identified, deep learning relies on training processes to find useful patterns in input examples. This approach makes training neural networks quicker and more efficient, leading to improved results in artificial intelligence. An algorithm is classified as deep if it undergoes a series of nonlinear transformations before producing output [26]. According to Kumar P. et al., recently, deep learning has proven to be good at recognizing actions and gestures, even better than other advanced methods. Deep learning is a part of machine learning (ML) that focuses on learning representations with multiple levels. It's like a tool within ML that can extract various levels of features. Especially in computer vision, deep learning has shown better performance than traditional ML methods. The key advantage of deep learning is that it can automatically learn features at different levels, capturing complex structures in the data without much manual effort. Deep learning uses a "deep" architecture because it processes information at different levels in a unique way, interpreting higher-level features in terms of lower-level features. This has led to improvements in creating powerful representations directly from raw data. Deep

learning is good at finding important hidden structures in data without labels and can be used for both pulling out features and sorting things into categories.

Once again, Deep Learning, a component of machine learning that relies on artificial neural networks for predictive analysis. Neural Networks involve nodes and statistical relationships between these nodes to model the way our minds work. It's called "Deep" because we're doing multiple layers of those neural networks. Each deep learning level is created with knowledge gained from the preceding layer of the hierarchy.

In 1943, Warren McCulloch and Walter Pitts proposed the initial artificial neuron model, laying the groundwork for neural networks and subsequent developments in deep learning. This was followed by Frank Rosenblatt's creation of the Perceptron algorithm in 1957, which stands as one of the earliest machine learning algorithms designed for pattern recognition. In 1967, J. A. Robinson invented the Resolution algorithm, serving as the cornerstone for automated theorem proving and logic-based machine learning. The year 1980 saw the advent of the backpropagation algorithm by Paul Werbos, enabling neural networks to learn from data by adjusting the weights of connections between neurons. Vladimir Vapnik and Alexey Chervonenkis contributed to machine learning advancements in 1995 with the development of the Support Vector Machine (SVM) algorithm, crucial for classification and regression tasks. Additional pioneers include Arthur Samuel, who coined the term "machine learning" in 1959, Geoffrey Hinton, recognized for significant contributions to deep learning in the 2000s, and Yann LeCun, who introduced the convolutional neural network (CNN) in the 1990s, now widely utilized in various computer vision applications.

Several neural network architectures, including radial basis function (RBF), counterpropagation, or learning vector quantization (LVQ) networks, are available for rapid prototyping [27]. While these architectures are relatively easy to train, they typically necessitate a large number of neurons, which is equivalent to the number of patterns or clusters in the dataset. Moreover, in many instances, these architectures mandate additional signal-normalization processes to achieve optimal performance.

1.3.1 Overview of deep learning frameworks

In 2024, the field of AI sees remarkable progress, with AI, ML, and DL frameworks emerging as indispensable tools shaping the creation, implementation, and deployment of intelligent systems. These frameworks, bolstered by extensive libraries and pre-built functions, empower developers to construct sophisticated AI algorithms efficiently, without needing to start from scratch. By streamlining the development process, they ensure consistency across projects and facilitate the integration of AI functionalities into diverse platforms and applications. Among the prominent frameworks, TensorFlow and PyTorch stand out as key players, offering a comprehensive suite of features spanning from machine learning to deep learning, thus catering to the evolving needs of research and development in the field of AI [28].

1. *TensorFlow*

TensorFlow, a formidable presence in the AI domain, was primarily developed by Google for machine learning and neural network research. Originating from Google's internal research in 2015 and evolving from the DistBelief framework, TensorFlow was designed to be more flexible and efficient. Its key features include a graph-based computation model, enabling efficient utilization of CPU and GPU resources, scalability for deployment across diverse computing environments, a versatile API catering to users of all skill levels, and TensorBoard [28] for visualization and debugging. With broad adoption in industry and academia, TensorFlow benefits from a large and active community of developers and researchers, solidifying its position as a powerhouse in the field of AI.

2. *PyTorch*

PyTorch, a state-of-the-art AI framework, is gaining traction within the machine learning and deep learning communities. Developed by Meta AI (formerly Facebook AI Research Lab) and built upon the Torch library, PyTorch made its debut in 2016, swiftly capturing attention due to its adaptability, user-friendly interface, and dynamic computation graph. Notable features include Autograd, facilitating dynamic adjustments during the learning process, and its deep integration with Python, enhancing accessibility for Python programmers. PyTorch boasts an extensive ecosystem encompassing libraries for computer vision (TorchVision) and natural language processing (TorchText), alongside robust support for GPU acceleration, rendering it suitable for high-performance model training and research. Bolstered by Meta's support and a vibrant

community, PyTorch continues to advance through contributions from academic researchers and industry professionals alike.

PyTorch and TensorFlow are two robust frameworks, each with distinct advantages. PyTorch is favored for research and dynamic projects due to its intuitive Pythonic approach, ideal for beginners and rapid prototyping. In contrast, TensorFlow excels in large-scale and production environments, offering optimized performance and scalability, especially in complex applications. While PyTorch provides flexibility for dynamic model adjustments, TensorFlow is becoming increasingly user-friendly with recent updates. In terms of community and resources, TensorFlow boasts a broad and established community, while PyTorch is rapidly gaining popularity, particularly in academic research circles. Real-world applications see PyTorch prevalent in academia and research-focused industries, while TensorFlow dominates in industry for large-scale deployments. Looking ahead, PyTorch focuses on usability enhancements, while TensorFlow prioritizes scalability and optimization, ensuring both frameworks continue to evolve to meet diverse user needs [28].

1.4 Key concepts

A machine learning model functions as a mathematical representation of a tangible process. Creating such a model entails supplying training data to a machine-learning algorithm, facilitating its comprehension of underlying patterns and relationships.

Training:

Training refers to the process of teaching a machine learning model to recognize patterns or make predictions based on input data. During training, the model learns the underlying patterns in the data by adjusting its parameters or weights through iterative optimization algorithms (e.g., gradient descent). The goal is to minimize the difference between the model's predictions and the actual outcomes in the training data.

A key factor to remember is that just because a model does well in training doesn't mean it will perform well in real-world situations. During training, the model learns from labeled data, but when it's put to the test with new, unseen data, it might not do as well. This is often because the model might have learned too much from the training data and struggles to adapt to new situations,

a problem called overfitting. To avoid this, a good learning algorithm needs to find a balance between minimizing errors during training and keeping things simple enough to work well with new data [29].

Testing:

Testing involves evaluating the performance of the trained model on data that it hasn't seen before, often referred to as the test dataset. The test dataset is separate from the training data and is used to assess how well the model generalizes to new, unseen examples. Testing helps determine the model's accuracy, precision, recall, and other performance metrics.

Validation:

Validation is a process used to fine-tune the parameters of a machine learning model and assess its generalization ability. It typically involves partitioning the dataset into three subsets: training, validation, and testing. The model is trained on the training dataset, and the validation dataset is used to tune hyperparameters or evaluate different model architectures. This iterative process helps prevent overfitting to the training data and ensures that the model performs well on unseen data.

Figure 8 below is a visual representation of how machines “learn”:

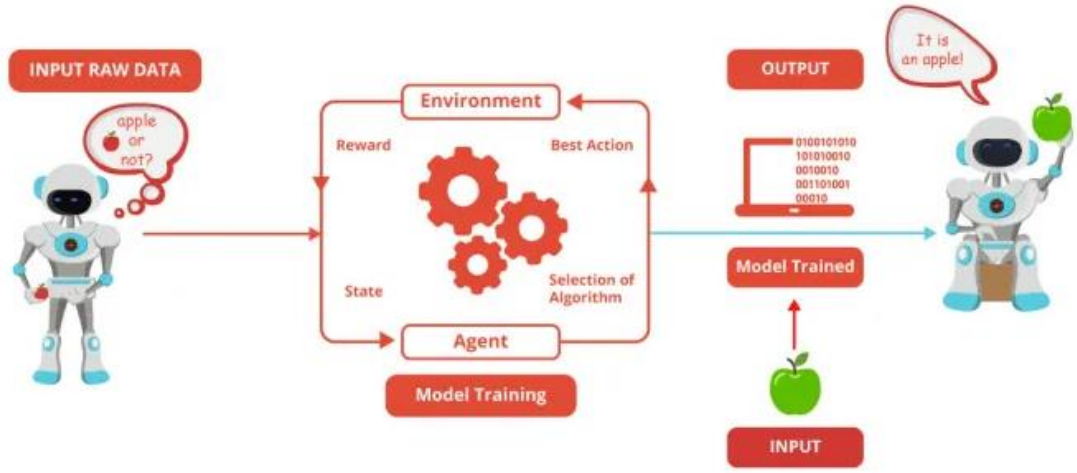


Figure 9: *Visual Representation of Machine Learning (Source: medium.com)*

Due to the relative nature of machine learning model performance, it is crucial to establish a robust reference point. A baseline represents a simple and widely accepted method for making predictions in your predictive modelling task. The effectiveness of this model sets the minimum acceptable

level of performance for a machine learning model tailored to your dataset. The outcomes generated by the baseline model act as the starting point against which the performance of all other models trained on your data can be evaluated.

Three examples of baseline models include:

1. Predicting the mean outcome value for a regression problem.
2. Determining the mode outcome value for a classification problem.
3. Forecasting the input as the output (referred to as persistence) for a univariate time series prediction task.

Consequently, the baseline performance in your particular context can serve as the standard against which all other models are compared and assessed.

If a model fails to achieve performance above the baseline, it suggests potential issues such as a software glitch or the model's unsuitability for your specific problem.

1.5 Overview

Chapter 1 provides a foundational understanding of artificial intelligence, covering its historical background and key concepts such as machine learning and deep learning. It introduces readers to different types of machine learning algorithms and the basics of neural networks. Pattern recognition is elucidated in chapter 2, emphasizing its significance in various domains. The chapter outlines different types of patterns and discusses the fundamentals of pattern recognition techniques, including statistical and structural approaches. Chapter 3 delves into the realm of gesture recognition, defining its importance and exploring existing technologies. It examines vision-based, sensor-based, and hybrid approaches to gesture recognition, along with associated challenges and applications across different fields. Gesture recognition systems enable intuitive interactions between humans and machines across various domains, including human-machine interaction, UAV rescue operations, and sign language recognition. This comprehensive overview provided in chapter 4 integrates insights from multiple studies focusing on vision-based, sensor-based, and hybrid approaches to gesture recognition. Chapter 5 provides a demonstration of key gesture recognition platforms such as Mediapipe, Leap Motion and Intel RealSense. It outlines

their significance, capabilities, and technical specifications, offering examples of applications developed using each platform, showcasing their versatility and potential in various domains. Chapter 6 explores the integration of gesture recognition technology to enhance interaction with UGVs. By leveraging intuitive gestures, we aim to streamline command input and improve operational efficiency. Finally, chapter 7 focuses on the discussion and future directions stemming from the findings presented in the thesis, as well as suggestions for further research.

CHAPTER 2: INTRODUCTION TO PATTERN RECOGNITION

2.1 Definition and concept of pattern recognition

Pattern recognition involves employing machine learning algorithms for the automatic identification of patterns and regularities within data. The data under consideration may encompass various forms such as text, images, sounds, or other discernible attributes. These systems exhibit the capability to recognize familiar patterns swiftly and precisely, as well as the ability to classify unfamiliar objects. Moreover, they can identify shapes and objects from diverse perspectives and discern patterns and objects even when partially obscured.

Pattern Recognition is the base. Everything around in this digital world is a pattern. We are basically surrounded by digital gadgets, so when one refers to digital world, everything around in this digital world is considered as a pattern. A pattern can either be seen physically or it can be observed mathematically by applying an algorithm.

Patterns can be observed in our day-to-day routine from the clothes we wear (identifying clothes by visualizing colors), to speech patterns whereas one can identify someone/something by the sound pattern. Everything we use has its own unique pattern.

To put it in a more direct concept [30], Pattern Recognition is a process of recognizing a pattern by using ML algorithms or in other words, classification of database on knowledge which was already gained from some statistical information. Patterns are kept in a knowledge base so that they can further contribute to performing classification, and for performing classification some ML algorithms are used. So, in a nutshell, whenever one uses ML algorithms for the purpose of classification, they'll basically need some prior or previously acquired patterns, helping the ML algorithm to work properly and as an output, given will be the classified object.

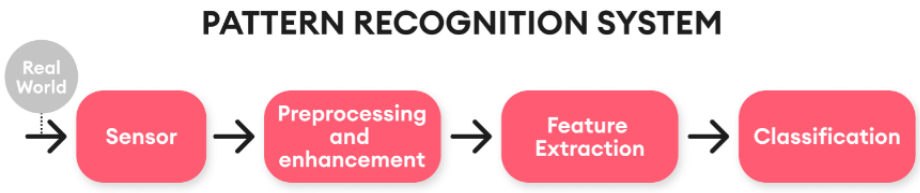


Figure 10: *The Pattern Recognition System (Source: superannotate.com)*

2.2 Brief historical overview

According to Bishop C. M. [30], finding patterns in data is a crucial task with a rich history of success. In the 16th century, Tycho Brahe's astronomical observations paved the way for Johannes Kepler to uncover planetary motion laws, contributing to classical mechanics. Likewise, identifying patterns in atomic spectra was vital for the development of quantum physics in the early 20th century.

The history of pattern recognition is intricately tied to the development of computing and our understanding of human cognition. Initially, pattern recognition was thought to be an ability unique to humans, but over time, it became apparent that computers could also recognize patterns. In the mid-20th century, there was a significant emphasis on patterns across various fields, including architecture and media. This sparked discussions about how both humans and computers recognize patterns and the similarities and differences between their approaches. Today, pattern recognition plays a crucial role in managing large amounts of information and understanding complex, interconnected systems in areas such as the environment, technology, and economics. It has played a pivotal role in shaping our modern digital landscape and continues to be a focal point of research and innovation [31].

2.3 Types of applications

Pattern recognition employs various techniques including statistical data analysis, probability, computational geometry, machine learning, and signal processing to derive insights from data. Given its widespread adoption across industries, the applications of this recognition model span diverse fields such as computer vision, object detection, speech, gesture and text recognition, and radar processing. In this section 3 types of applications will be presented including image, speech, and gesture along with some examples of real-world applications.

2.3.1 Image Patterns

- *Characteristics:*[32] Image patterns involve the analysis and recognition of visual data, such as photographs, diagrams, or scanned documents. This encompasses tasks like object recognition, scene understanding, and image classification.

- *Challenges*: Challenges in image pattern recognition include variability in lighting conditions, occlusions, viewpoint changes, and background clutter. Additionally, handling large-scale datasets and achieving real-time performance are significant challenges.

2.3.2 Speech Patterns

- *Characteristics*: Speech patterns involve the interpretation and recognition of spoken language. This includes tasks such as speech recognition, speaker identification, and emotion detection from speech signals.
- *Challenges*: Challenges in speech pattern recognition include dealing with variations in accents, speech rates, background noise, and speech disorders. Achieving robustness in noisy environments and handling diverse languages and dialects are also major challenges.

2.3.3 Gesture Patterns

- *Characteristics*: Gesture patterns pertain to the interpretation and recognition of human gestures and body movements. This includes tasks like gesture recognition in human-computer interaction systems, sign language recognition, and action recognition in videos.
- *Challenges*: Challenges in gesture pattern recognition include capturing and interpreting complex and subtle movements accurately. Additionally, dealing with occlusions, viewpoint changes, and variations in gesture styles across individuals poses significant challenges. Integrating multimodal information, such as combining visual and depth data for gesture recognition, also presents technical hurdles [33].

2.4 Real-world Applications of Patterns

Real-world applications of patterns are found in many everyday technologies, making our lives easier and more efficient. For instance, smartphones use pattern recognition to suggest words as you type or to organize photos by recognizing faces and places. Streaming services like Netflix and Spotify analyze your viewing and listening habits to recommend movies and music you might enjoy. Email services use patterns to filter out spam and prioritize important messages. Even in home appliances, smart thermostats learn your heating and cooling preferences to optimize energy usage. These simple examples show how pattern recognition seamlessly integrates into our daily

routines, enhancing convenience and personalization. Below are presented some applications on various sectors.

2.4.1 Image Pattern Applications

Image pattern recognition has become an integral part of various industries, enhancing capabilities and improving efficiency. In healthcare, it plays a crucial role in medical imaging by detecting tumors in MRI or CT scans, identifying anomalies in X-rays, and segmenting organs for accurate diagnosis. In the transportation sector, autonomous vehicles rely on pattern recognition for lane detection, traffic sign recognition, pedestrian detection, and obstacle avoidance, ensuring safer and more efficient travel. The retail industry leverages pattern recognition for customer behavior analysis, enabling businesses to analyze purchasing patterns and preferences to create targeted marketing campaigns and personalized recommendations. Security and surveillance systems use facial recognition technology to enhance security in public places, airports, and border control. Additionally, pattern recognition is pivotal in manufacturing for quality control, as it detects defects in products and ensures high product quality through advanced image analysis.

2.4.2 Speech Patterns Applications

Speech pattern recognition has numerous applications across various sectors. In healthcare, it enhances patient identification and access control through biometric authentication via voice recognition. In the finance industry, speech pattern recognition aids in fraud detection by analyzing voice patterns to detect fraudulent activities in financial transactions. In transportation, it plays a crucial role in transportation planning by analyzing traffic patterns and optimizing routes and schedules based on spoken instructions or queries.

2.4.3 Gesture Patterns Applications

Gesture pattern recognition has a wide range of applications in different fields. In transportation, it enables gesture-based interaction for controlling vehicle functions or providing navigation instructions in autonomous vehicles. In the retail sector, analyzing gestures and body language helps understand customer engagement and satisfaction levels, enhancing customer behavior analysis. In security and surveillance, gesture pattern recognition is used for intrusion detection by

analyzing body movements to identify suspicious behaviors or unauthorized access in restricted areas[32], [33], [34].

2.5 Overview of pattern recognition techniques

In pattern recognition, selecting the right algorithms is a significant challenge, given the multitude of options available. Here, we'll briefly outline five common types of algorithms used in recognition:

1. ***Statistical***

This approach relies on probability and utilizes statistical techniques to learn from examples. By analyzing observations, the model deduces rules that can be applied to future data. Statistical pattern recognition primarily focuses on accurately categorizing patterns into predefined classes. Essential components of statistical pattern recognition encompass various stages, including pre-processing, feature extraction, and selection. This is followed by the estimation of probability densities, which can be done parametrically or non-parametrically. Decision-making processes are then employed to assign patterns to appropriate classes, with subsequent evaluation of performance and potential post-processing adjustments. The learning process may involve supervised or unsupervised methods, along with cluster analysis to identify inherent patterns within the data. Overall, statistical pattern recognition involves a comprehensive approach to effectively classify patterns and derive meaningful insights from data [35].

2. ***Structural***

Unlike statistical methods, structural recognition employs a hierarchical approach, categorizing patterns into subclasses. It describes complex relationships between elements and is suitable for tasks like image and shape analysis.

The concept of structural pattern recognition was first introduced by Pavilidis [36] in 1977. Unlike traditional methods relying on segmentation and feature extraction, structural pattern recognition focuses on describing the arrangement of sub-patterns within a larger pattern. It aims to explain how simple sub-patterns combine to form complex patterns. There are two main approaches in structural pattern recognition: syntax analysis and structure matching. Syntax analysis is rooted in

formal language theory, while structure matching employs mathematical techniques to analyze sub-patterns. Structural pattern recognition excels in capturing relationships among different parts of an object and is particularly effective in handling symbolic information, making it suitable for applications requiring higher-level analysis such as image interpretation.

Furthermore, structural pattern recognition often integrates with statistical classification or neural networks to tackle more complex pattern recognition problems, including the recognition of multidimensional objects. This integration enhances the capability of structural pattern recognition systems to address diverse and challenging pattern recognition tasks.

3. Neural Network

Neural networks, inspired by biological concepts, are highly flexible for pattern recognition. Feed-forward networks, in particular, excel in classification by learning from input patterns through feedback. Neural networks have undergone rapid development since the proposal of the first model, the McCulloch-Pitts (MP) neuron, in 1943. Notably, the Hopfield neural networks and the widely acclaimed backpropagation (BP) algorithm have significantly advanced the field. Neural networks are a data clustering method that relies on distance measurement and is independent of specific models. [35]Drawing inspiration from biological concepts, neural networks mimic the human brain's physiology to recognize patterns. Comprising a series of interconnected units, neural networks leverage genetic algorithms, a statistical optimization technique introduced by Holland in 1975. Neural Pattern Recognition (NeurPR) stands out for its minimal need for prior knowledge. With sufficient layers and neurons, an artificial neural network (ANN) can effectively create complex decision regions, making it an attractive tool for pattern recognition tasks.

4. Fuzzy-based

Fuzzy logic, reflecting the uncertainty inherent in real-world recognition tasks, is utilized in this algorithm. It's particularly useful when dealing with uncertain components in visual recognition. Human thinking processes often involve uncertainty and fuzziness, reflected in the language we use. However, reality is often complex, and complete answers or classifications may not always be feasible. To address this, the theory of fuzzy sets emerged, offering a framework to effectively describe the extension and intension of concepts. The application of fuzzy sets in pattern recognition dates back to 1966, with Bellan et al. focusing on abstraction and generalization.

Principles proposed by Marr (1982) and Keller (1995) further underscore the importance of fuzzy sets in pattern recognition. A pattern recognition system based on fuzzy sets theory [37] has the capability to closely mimic the human thinking process, providing a versatile and deep approach to pattern recognition.

5. Hybrid

Hybrid models combine different algorithms to leverage their respective advantages. These models employ multiple classifiers trained on feature spaces, with a decision function determining the final output based on the accumulated classifier sets' accuracy [38].

CHAPTER 3: LITERATURE REVIEW ON GESTURE RECOGNITION

3.1 Definition and significance of gesture recognition

According to Ouda M. et. Al. [39], the main objective in researching gesture recognition is to develop a system capable of identifying particular human gestures and utilizing them for communication or control purposes. This entails not only tracking human movements but also interpreting those movements as meaningful commands. Typically, two approaches are employed for interpreting gestures in human-computer interaction (HCI) applications. The first approach utilizes data gloves, either wearable or in direct contact with the hand. The second approach relies on computer vision, eliminating the necessity for wearing any sensors.

Hand gestures encompass elements of non-verbal communication expressed through the orientation of the palm, positioning of fingers, and configuration of the hand. They can be categorized into static gestures, which maintain a fixed hand shape, and dynamic gestures, involving a sequence of hand movements like waving. Each gesture comprises various hand motions, such as the unique variations in handshakes between individuals and across contexts. Unlike posture, which primarily concerns the static form of the hand, gestures emphasize hand movement. Research on hand gestures typically falls into two primary methodologies: the wearable glove-based sensor approach and the camera vision-based sensor approach.

Simply put by Choudhury A. et. Al. [40], gesture recognition facilitates real-time communication between a user and a computer by interpreting hand and body movements as commands. Typically, motion sensors within a device capture and interpret gestures, serving as the primary means of input. Many gesture recognition systems incorporate 3D depth-sensing cameras and infrared cameras alongside machine learning algorithms. These algorithms are trained using labeled depth images of hands, enabling them to identify hand and finger positions accurately.

Gesture recognition involves three fundamental stages as also seen on figure 11 below:

- *Detection*: Using a camera, the device identifies hand or body movements. A machine learning algorithm then analyzes the image to locate hand edges and positions.

- *Tracking*: The device monitors movements frame by frame to capture each gesture accurately, providing precise input for data analysis.
- *Recognition*: The system analyzes the gathered data to identify patterns. Upon detecting a match and interpreting a gesture, it executes the associated action. This recognition functionality is implemented through feature extraction and classification, as depicted in the provided scheme.

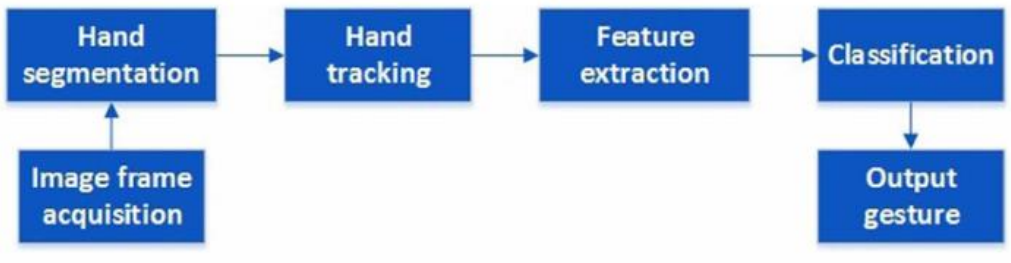


Figure 11: *Fundamental stages of Gesture Recognition (Source: Research Gate)*

According to Grand View Research Magazine, in 2022, the market size for gesture recognition reached USD 17.29 billion and is forecasted to witness a compound annual growth rate (CAGR) of 18.8% from 2023 to 2030. Factors contributing to this growth include the global rise in per capita incomes, ongoing technological innovations, and the expanding digitization observed across various sectors like automotive, consumer electronics, and healthcare. Moreover, the escalating adoption of consumer electronics, the proliferation of the Internet of Things (IoT), and the growing demand for enhanced user comfort and convenience further propel the market's expansion.

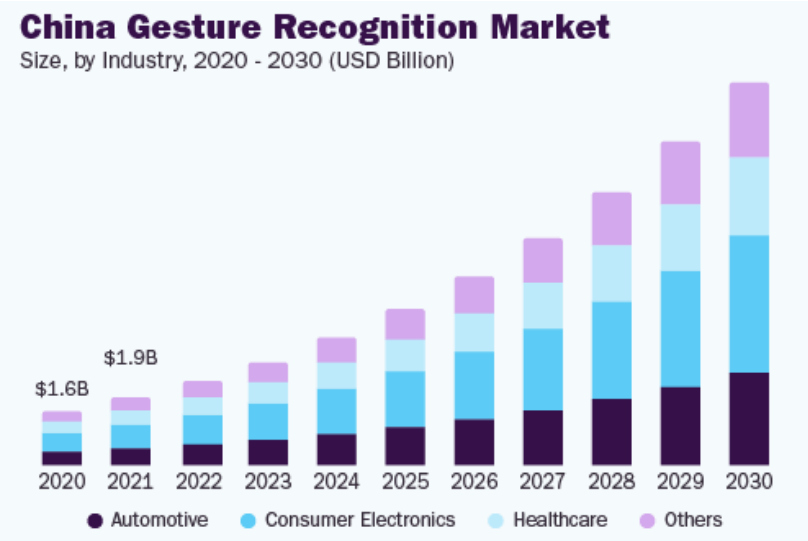


Figure 12*: China Gesture Recognition Market (Source: Grand View Research)*

3.2 Brief history of gesture recognition

Since the inception of machines, humans have strived for them to operate according to human intentions. In the early stages of machine development, people utilized buttons, joysticks, and other controls to convey orders to the machine, manipulating circuits, oil, and mechanical transmission. With the advent of computers in modern times, the human-machine interface (HMI) has become increasingly user-friendly. Individuals can now interact with machines through devices like mice and keyboards, while monitoring their operations via displays. In recent years, computers have drastically reduced in size, leading to significant enhancements in machine efficiency.

Hand gestures present an intriguing area of study due to their potential to enhance communication and offer an instinctive mode of interaction applicable across diverse contexts.

In the past, hand gesture recognition relied on wearable sensors integrated into gloves, which detected physical responses to hand movements or finger bending. Subsequently, the collected data were processed using a wired connection to a computer. This glove-based sensor system could be rendered portable by utilizing sensors connected to a microcontroller.

The history of hand gestures in human-computer interaction (HCI) traces back to the development of data glove sensors, which provided basic commands for interfacing with computers. These gloves employed various sensor types to capture hand motion and position accurately by determining the precise coordinates of the palm and fingers. Among the sensors that utilized a similar bending angle technique were curvature sensors, angular displacement sensors, optical fiber transducers, flex sensors, and accelerometer sensors, each exploiting distinct physical principles [31].

Elderly individuals grappling with chronic conditions causing muscle loss may find it challenging to wear or remove gloves, leading to discomfort and mobility constraints, particularly during extended use. Moreover, individuals with sensitive skin or those recovering from burns may experience skin damage, infections, or adverse reactions from sensors. Additionally, the expense of certain sensors poses a barrier.

Addressing these concerns, Lamberti and Camastra [41] introduced a computer vision system utilizing colored marked gloves, eliminating the need for attached sensors but still requiring the use of gloves.

These limitations prompted the development of innovative and cost-efficient techniques that eliminate the need for cumbersome gloves. Known as camera vision-based sensor technologies, these methods leverage advancements in open-source software libraries, facilitating the detection of hand gestures across various applications such as clinical operations, sign language, robot control, virtual environments, home automation, personal computers, tablets, and gaming. These techniques involve replacing instrumented gloves with cameras, including RGB, time-of-flight (TOF), thermal, or night vision cameras.

Algorithms utilizing computer vision methods have been devised to detect hands using these camera types. These algorithms aim to segment and identify hand features, including skin color, appearance, motion, skeleton, depth, 3D models, deep learning detection, among others [31].

3.3 Overview of existing technologies

Studies indicate that technology employing hand gestures can be categorized into three types: sensor-driven, vision-driven, and hybrid approach. Sensor-based technology utilizes various sensors like accelerometers and gyroscopes. On the other hand, [42][3] vision-driven technology relies on RGB cameras and infrared sensors to extract and identify features from datasets of hand movements.

3.3.1 Sensor-based approach

Sensor-based hand gesture recognition algorithms utilize motion sensors, which are either embedded within gloves or employed in smart devices like smartphones. These sensors typically include built-in accelerometers along with gyroscope sensors.

Al Farid F. et. Al. [42] imply that the wearable glove-based sensors can be used to capture hand motion and position. In addition, they can easily provide the exact coordinates of palm and finger locations, orientation and configurations by using sensors attached to the gloves. However, this approach requires the user to be connected to the computer physically, which blocks the ease of interaction between user and computer. In addition, the price of these devices is quite high. However, the modern glove-based approach uses the technology of touch, which more promising technology and it is considered Industrial-grade haptic technology. Where the glove gives haptic

feedback that makes the user sense the shape, texture, movement and weight of a virtual object by using microfluidic technology. Image 2 shows an example of a sensor glove used in sign language.

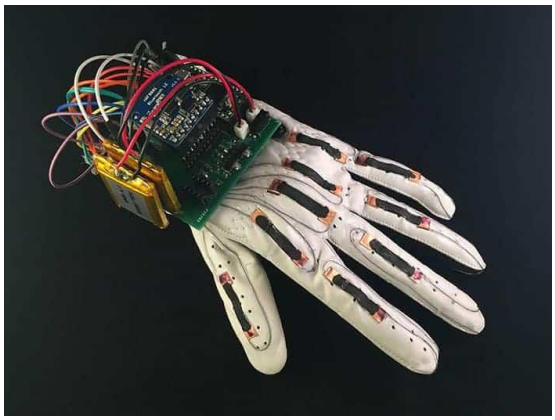


Image 2: *Sensor-based data glove (Source: physicsworld.com)*

3.3.2 Vision-based approach

The vision-based method comprises three primary phases: image acquisition, image segmentation, and image identification. Numerous scholars have devised real-time hand motion detection systems based on these stages. Dynamic recognition of hand gestures involves identifying a moving hand with various motions, while static hand gestures entail recognizing hand positions. Hand motions can be visually classified using methods like 3D modeling and appearance-based approaches. These include visual sub-models of 3D models and model-based appearance methods. The 3D hand gesture model offers a spatial representation of the human hand over time. Appearance-based hand gesture representation methods encompass four types: color, silhouette, texture, and motion-based models.

The camera vision-based sensor is widely recognized as a practical and versatile technique because it enables non-contact communication between humans and computers. Various camera configurations can be employed, including monocular, fisheye, time-of-flight (TOF), and infrared (IR) cameras. Nonetheless, employing this technique poses several challenges, such as variations in lighting conditions, background interference, the impact of occlusions, complex backgrounds, the trade-off between processing time, resolution, and frame rate, and instances where foreground or background objects may have similar skin color tones or resemble hands in appearance [42].

3.3.2.1 Color-Based Recognition

A) Utilizing Glove Marker for Color-Based Recognition

This technique employs a camera to monitor hand movements by utilizing a glove marked with various colors, depicted in Image 3.

***Image 3**: Color-based recognition using colored glove (Source: [43])*

It has been employed for interacting with 3D models, enabling actions such as zooming, moving, drawing, and using a virtual keyboard with considerable flexibility. The colors on the glove facilitate the camera sensor in tracking and identifying the palm and finger positions, allowing for the extraction of a geometric model representing the hand's shape. This method offers simplicity and affordability compared to sensor-based data gloves. Nonetheless [43], it still necessitates wearing colored gloves and may restrict the level of natural and spontaneous interaction with the human-computer interface.

B) Color-Based Recognition of Skin Color

Skin color detection is a widely used method for hand segmentation, finding applications in diverse areas such as object classification, image enhancement, movement tracking, video surveillance, human-computer interaction (HCI), facial recognition, and gesture identification. It can be achieved through two main approaches: pixel-based and region-based skin detection. Various color spaces, including RGB, HSV, and YCbCr, are commonly utilized to represent image color information, each offering advantages and disadvantages depending on the application. For instance, HSV is preferred for its resilience to lighting variations, while YCbCr simplifies the process of skin tone detection.

[44]However, skin color detection methods face challenges such as illumination variations, background interference, and noise. Several studies have addressed these challenges through innovative approaches, including combining frame differencing and skin color segmentation,

motion-based segmentation, cross-correlation, and hybrid methods integrating histogram analysis and Gaussian mixture models.

Despite advancements, challenges persist, such as sensitivity to background elements matching skin color and limitations in low-light conditions. Researchers continue to explore novel techniques, such as integrating multiple cameras or leveraging depth information, to enhance skin color detection and overcome existing limitations in various applications [42].

3.3.2.2 Appearance-Based Recognition

This technique extracts image features directly from pixel intensities without segmentation. It employs methods like Haar-like features and AdaBoost algorithm for efficient pattern recognition, enabling real-time execution and detection of various skin tones [45].

***Image 4**: Foreground extraction is used in appearance recognition to isolate the Region of Interest (ROI), utilizing techniques such as pattern subtraction and foreground segmentation algorithms.*

3.3.2.3. Motion-Based Recognition

Utilizing motion patterns extracted from image frames for gesture recognition, this approach often involves optical flow computation and color space analysis. [39] Machine learning algorithms like boosting are commonly used for classification, ensuring robust detection despite dynamic backgrounds and occlusion.

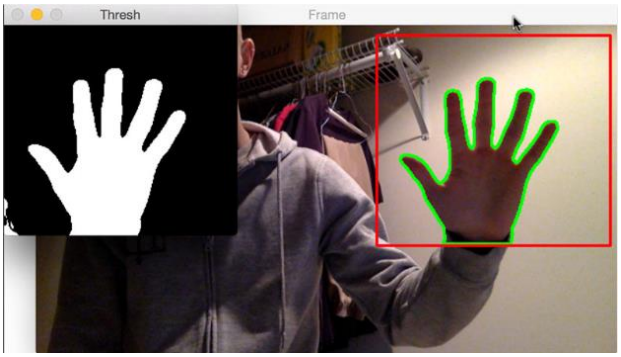


***Image 5**: On the left is the isolated hand extracted from the Region of Interest (ROI) using thresholding. On the right, contours have been drawn around the hand.*

3.3.2.4 Skeleton-Based Recognition

Involves modeling hand skeletons to enhance detection accuracy. Algorithms focus on joint orientation, trajectories, and geometric attributes, often utilizing depth sensor data combined with skeletal information for improved segmentation and tracking [17] [39].

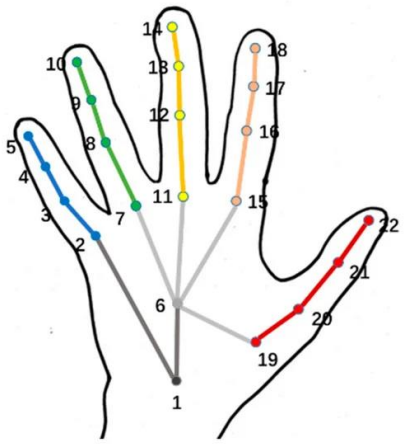


***Image 6**: Twenty-two joints of a right-hand skeleton*

3.3.2.5 Depth-Based Recognition

This method leverages depth information from cameras to provide 3D geometric data. Techniques include depth threshold segmentation and combining depth with color information for robust hand segmentation. Classification is achieved using algorithms like kNN classifiers and depth thresholding.

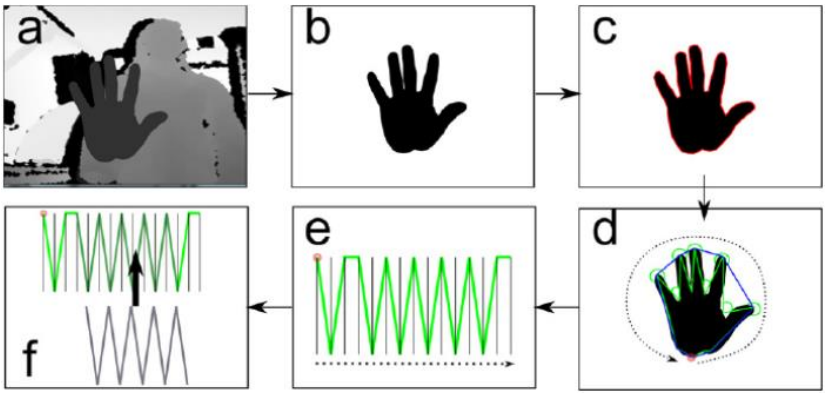


***Image 7:** Hand gesture detection process on depth images: (a) Original depth image. (b) Segmentation of closest objects through thresholding. (c) Blob detection and contour extraction (red). (d) Polygonal approximation of contours (green), determination of convex*

3.3.2.6 3D Model-Based Recognition

Relies on 3D kinematic hand models to estimate hand poses and interactions. Techniques range from deep learning models for single RGB or depth images to methods for tracking hand movement in 3D space using advanced algorithms like the Hungarian algorithm for point matching [31].

3.3.2.7 Event-based Recognition

Event-driven sensor-based recognition is a technology that processes and responds to data collected by sensors in real-time, based on specific triggers or events. This approach is widely used in smart home systems, where sensors detect changes in the environment—such as motion, temperature, or light—and trigger appropriate actions, like turning on lights or adjusting thermostats. In security systems, sensors can detect unauthorized entry or unusual activities, instantly alerting homeowners or authorities. Event-driven recognition is also pivotal in industrial automation, where sensors monitor machinery and processes, initiating maintenance or adjustments to prevent failures and enhance efficiency. By responding dynamically to real-time data, event-driven sensor-based recognition significantly enhances the functionality and responsiveness of modern systems.

Jolley A. explores biologically inspired, or neuromorphic, event-based sensors that offer microsecond temporal resolution and an exceptionally high dynamic range. He highlights their use in space surveillance, satellite tracking, and characterization, providing real-life applications of event-based recognition [46].

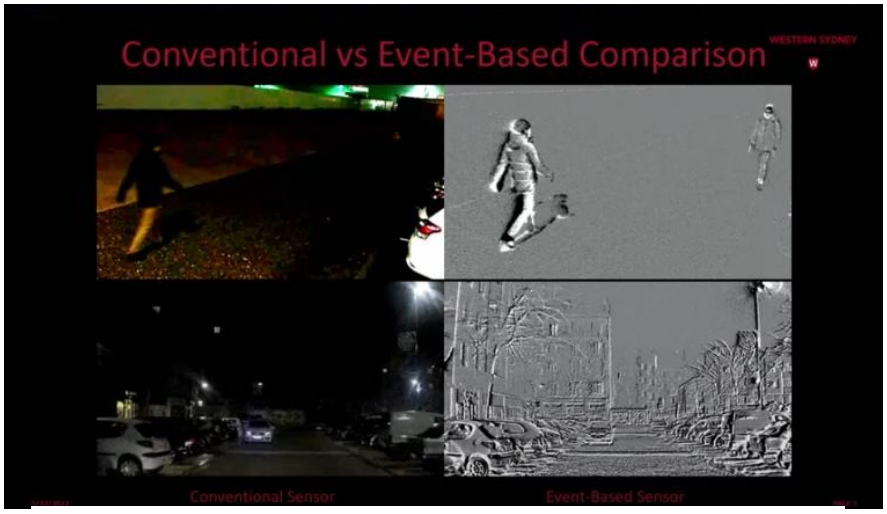


Image 8: Conventional vs Event-Based Comparison (Source:[46])

Image 8 above is an example of event-based recognition. With the conventional sensor, we can't distinguish much detail about the buildings in the background, which you can clearly see with the event-based sensor. Additionally, the bright lights that saturate the pixels in the conventional sensor don't affect the event-based sensor as significantly.

3.3.3 Hybrid approach

Artificial intelligence provides a robust and dependable approach utilized across various modern applications due to its utilization of the learning-by-example principle. Deep learning, with its multilayered architecture, excels in learning from data and delivering accurate predictions. However, one of the primary challenges associated with this technique is the need for extensive datasets to train the algorithm, which can significantly impact processing time.

Gesture recognition technologies have undergone significant advancements in recent years, fueled by breakthroughs in computer vision, machine learning, and sensor technologies. These innovations have led to the development of various approaches to gesture recognition, each leveraging different technologies for unique applications. Camera-based gesture recognition, exemplified by Microsoft Kinect and Intel RealSense, employs RGB, depth-sensing, or infrared cameras to capture and interpret hand or body movements. Depth sensing technologies, including LiDAR and Time-of-Flight sensors, enable precise 3D motion tracking by measuring distances between objects and the sensor. Electromyography (EMG) technology detects muscle contractions' electrical activity, ideal for interpreting gestures based on muscle movements, particularly in prosthetics and wearables. Wearable devices like smartwatches and AR/VR headsets integrate sensors such as accelerometers and gyroscopes to detect wearer gestures for navigation and interaction. Radar-based systems, ultrasonic sensors, inertial measurement units (IMUs), and glove-based systems offer additional avenues for gesture recognition, each with distinct functionalities and applications. Machine learning and AI algorithms play a crucial role in analyzing and interpreting gesture data captured by these sensors, enhancing accuracy and robustness. By combining or selectively utilizing these technologies, gesture recognition systems can cater to diverse application requirements, considering factors like accuracy, real-time processing, environmental conditions, and cost constraints.

3.4 Applications in various fields

3.4.1 Clinical and Health

During medical procedures, surgeons often require comprehensive insights into a patient's anatomy or detailed organ models to enhance procedural efficiency and precision. Medical imaging technologies such as MRI, CT scans, or X-rays [47] capture intricate details from the patient's body, presenting them as high-resolution images on a screen. Surgeons can manipulate these images using hand gestures detected through computer vision techniques. These gestures enable them to perform various tasks like zooming, rotating, cropping images, or navigating through slides, all without the need for additional peripherals such as mice, keyboards, or touch screens. By minimizing the use of extra equipment, the need for sterilization is reduced, particularly for devices like keyboards and touch screens. Furthermore, hand gestures can also serve assistive functions, such as controlling wheelchairs [39]. HGR technology, as demonstrated by Microsoft, enables accessing information and imaging during surgical procedures or other manipulations. GestSure, a company specializing in this technology, empowers doctors by allowing them to check MRI, CT scans, and other imagery using simple gestures, eliminating the need to scrub out. This innovation streamlines the workflow for medical professionals, providing convenient and efficient access to critical medical imaging data without disrupting the surgical process.

***Image 9**: Application of Gesture Recognition in Healthcare (Source: www.gestsure.com)*

3.4.2 Sign Language Recognition

Sign language serves as an alternative communication method for individuals who cannot speak. It comprises a series of gestures, with each gesture representing a letter, number, or expression. Various research papers have explored sign language recognition for deaf individuals, often employing either a glove-attached sensor or camera-based computer vision techniques. These methods capture hand movements and interpret them into responses. In both approaches, the dataset used for gesture classification aligns with real-time gestures made by the user [39].

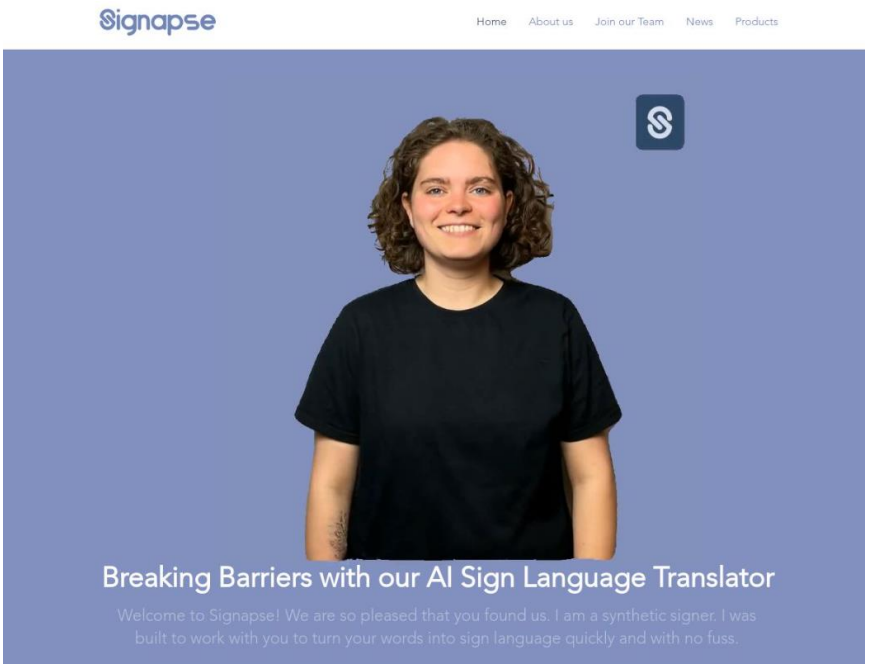


***Image 10**: Source: Application of Gesture Recognition in Sign Language (https://www.signapse.ai/)*

An actual application is 'Signapse's Sign Language Translation using Generative AI'. Signapse's implementation of Generative AI for sign language translation marks a significant advancement in providing realistic and accurate interpretations for the d/Deaf community. Leveraging Generative Adversarial Networks (GANs) and deep learning techniques, Signapse delivers unparalleled precision and lifelike representations in its sign language translations. Utilizing Generative Adversarial Networks (GANs), Signapse pioneers the creation of photo-realistic sign language videos. This approach offers distinct advantages, including heightened realism and a more immersive experience for d/Deaf users compared to conventional Motion Capture-based avatars. The emphasis on accuracy in sign language translation is paramount, ensuring that d/Deaf individuals can comprehensively grasp the conveyed information without discrepancies or misunderstandings. Accurate translations are not only essential for conveying precise meaning but also for respecting the nuances and cultural aspects inherent in sign language.

Signapse's [48] strength lies in its commitment to accuracy, facilitated by extensive datasets of sign language videos curated by proficient translators. Through continuous updates and enhancements, Signapse ensures the relevancy and clarity of its translations, keeping pace with evolving sign language conventions and vocabulary. The translation process itself involves meticulous selection of pre-recorded videos, blending them seamlessly through AI-powered algorithms, and providing real-time updates for dynamic announcements in public spaces.

3.4.3 Robot Control

Robot technology is utilized across a multitude of domains, spanning industry, assistive services, retail, sports, and entertainment. Advanced robotic control systems employ machine learning, artificial intelligence, and intricate algorithms to carry out designated tasks, empowering robots to engage naturally with their surroundings and autonomously make decisions. Research endeavors explore the integration of computer vision technology with robots to create assistive solutions for the elderly population. Additionally, studies focus on leveraging computer vision to enable robots to seek guidance from humans for effective navigation within specific buildings [49]. A real-life example is ADAM [50] **.**

Image 11: *Source: www.researchgate.net*

The collaboration between Universidad Carlos III de Madrid and Robotnik has led to the creation of ADAM, a versatile personal robot designed to support the elderly population. ADAM employs modular design and advanced systems to perform household tasks, making it ideal for indoor environments. Equipped with vision and gripper systems, ADAM adapts to home environments, enhancing the quality of life for its users by carrying out various physical tasks.

It fosters a collaborative learning environment while respecting personal space, positioning itself as a genuine companion for those in need. ADAM's modular composition allows for both independent and integrated operation, with batteries powering its base and arms for autonomy within the home. Integrated sensors prevent collisions and optimize task

performance, while its ability to learn new skills through interactions sets a new standard in personalized robotic assistance.

Implemented in home settings, ADAM has proven effective, particularly in tasks like cleaning, simplifying daily routines and promoting independence and well-being. Further testing is needed to validate its effectiveness and acceptance among its target audience.

Future development will focus on hardware and software improvements to expand ADAM's capabilities in vision, handling, and navigation. These advancements will solidify ADAM's position as an indispensable assistant for elderly care and as a trailblazer in personal robotic assistance.

3.4.4 Gaming

One of the most well-known apps that utilize gesture recognition in gaming is "Kinect Sports" for the Xbox gaming console. Kinect Sports utilizes the Kinect sensor, which enables players to control gameplay using body movements and gestures rather than traditional controllers. The game features various sports activities such as soccer, bowling, tennis, and boxing, where players can kick, swing, and punch to interact with the game. This innovative use of gesture recognition technology provides an immersive and interactive gaming experience, making Kinect Sports a popular choice among gamers.

Kinect Sports for the Xbox 360 introduces players to a lively world of cartoonish avatars engaging in a variety of sports activities, creating a cheerful and approachable atmosphere. Unlike other video games, Kinect Sports revolutionizes gameplay by incorporating full-body movement tracking, emphasizing physical activity beyond simply waving arms. Whether it's track and field events or bowling, players find themselves physically moving within Kinect's play space, requiring genuine exertion and immersion. The game's immersive nature challenges players to perform real-life actions, resulting in a workout-like experience that leaves players sweaty and sore, underscoring its "Very Active" label [51].

Image 12: *Application of Gesture Recognition in Gaming (Source:[51])*

[52]

3.4.5 Space

A great example of event-based recognition is the Falcon Neural Satellite, a partnership between Western Sydney University and the US Air Force Academy. Falcon Neuro is an experiment currently flying on the International Space Station (ISS), designed and built by cadets and faculty at the United States Air Force Academy (USAFA). It marks the first use of biologically inspired event-based, or neuromorphic, cameras in space. Despite its small size, Falcon Neuro is quite powerful, containing two neuromorphic cameras that were modified for space use by Western Sydney University (WSU) in Sydney, Australia. One camera is positioned to look downward (Nadir) and the other forward (Ram), both controlled by electronics developed at the Space Physics and Atmospheric Research Center (SPARC) in the Department of Physics and Meteorology at USAFA [53].

The Department of Defense Space Test Program (STP) facilitated Falcon Neuro's journey to the ISS, where it began operations in January 2022, continuing until January 2024. Managed from a dedicated ground station at USAFA, Falcon Neuro operates most weekdays and has captured over 200 recordings to date.

The sensor is placed on the satellite as seen in image 13 below as seen circled in red.

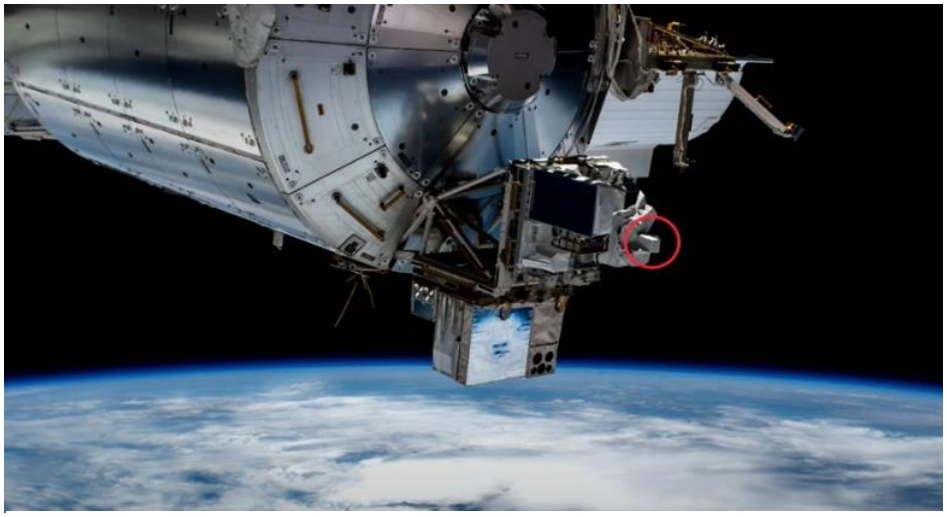


Image 13: Falcon Neuro (Source:afresearchlab.com)

Neuromorphic cameras are innovative tools for space-based imagery, offering significant advantages by reading out much less data much more quickly, enabling hyper-temporal (faster-than-real-time) recording capabilities. This is crucial for detecting challenging threats such as hypersonic re-entry vehicles, missiles, or fast-moving aircraft. At the USAFA, future Air and Space Force leaders are gaining firsthand experience in space technology through the Falcon Neuro experiment. This initiative has allowed cadets from various departments, including Physics and Meteorology, Astronautics, Electrical, Mechanical, Computer, Aero, and Military Strategic Studies, to engage with cutting-edge space technology by participating in the building, testing, and operation of the Neuro system[53].

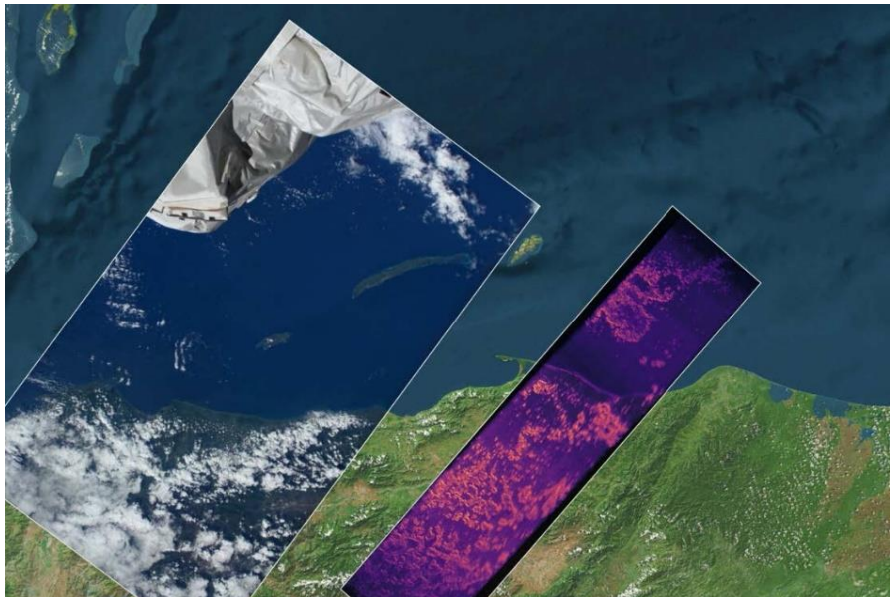


Image 14:Comparison between the NASA ISS HD camera and the motion-compensated image of Honduras captured by Neuro. (Source: United States Air Force Academy and Western Sydney University)

CHAPTER 4: OVERVIEW OF EXISTING TECHNOLOGIES ON GESTURE RECOGNITION

4.1 Vision-Based Gesture Recognition

Vision-based approaches leverage image processing, machine learning, and depth sensing technologies to interpret gestures. These methodologies include appearance-based recognition, motion-based recognition, skeleton-based recognition, depth-based recognition, and 3D model-based recognition. Algorithms such as AdaBoost, Optical Flow, SVM, and CNNs are employed for feature extraction, classification, and pose estimation, enabling natural interaction with devices across various domains [33], [39].

4.1.1 Technology Overview

Vision-based gesture recognition utilizes cameras or depth sensors to capture images or depth maps of hand movements. These inputs are processed to extract relevant features and classify gestures using computer vision techniques.

The methodologies used were Appearance-based, Motion-based, Skeleton-based, Depth-based and 3D model-based recognition (see analytically in Chapter3).

Technologies used in the studies were Image processing, Machine learning and Depth sensing and more specifically:

- Cameras: Utilized for capturing images or videos of hand gestures.
- Depth Sensors: Used to capture depth information for 3D gesture recognition.
- Image Processing: Techniques applied to preprocess images or videos for feature extraction and analysis [39].

4.1.2 Algorithms

Several advanced techniques are employed in gesture recognition systems. Haar-like Features and AdaBoost are utilized for appearance-based recognition, focusing on efficient feature selection and classification [39]. Optical Flow is implemented to track motion patterns within video sequences, while the CAMShift Algorithm analyzes color spaces to detect gestures. Gesture classification is

further enhanced using Histogram Distribution Models based on extracted features. Additionally, Graph Convolutional Neural Networks (Graph CNN) play a crucial role in reconstructing 3D mesh hand surfaces and estimating both 3D hand shapes and poses. These methods collectively enable robust and accurate gesture recognition across various applications.[39]

4.1.3 Applications

1.Human-Computer Interaction (HCI):

Gesture-based control extends beyond traditional input methods, enabling users to navigate interfaces, select options, and perform actions on computers, tablets, and smartphones without physical touch, as detailed in [39]. This technology also facilitates touchless interaction in public spaces like airports and hospitals, reducing the transmission of germs and enhancing user convenience. These applications leverage vision-based recognition methodologies such as appearance-based, motion-based, and depth-based recognition to accurately interpret and respond to gestures in real-time scenarios[39].

2.Virtual and Augmented Reality (VR/AR):

Gesture-based interaction plays a pivotal role in virtual environments, enabling intuitive manipulation of virtual objects and immersive experiences in gaming, training simulations, and other applications[33], [54]. Additionally, in industrial settings, augmented reality applications leverage hand gestures for accessing information overlays, performing maintenance tasks, and operating machinery hands-free, as highlighted in [34]. These technologies rely on vision-based recognition systems to accurately track and interpret hand gestures within both virtual and augmented environments, enhancing user interaction and operational efficiency.

3.Public Safety and Security:

Gesture recognition technology serves diverse applications in enhancing security and emergency response scenarios. In surveillance systems, it facilitates the identification of suspicious behaviors and potential security threats in public spaces, providing proactive monitoring and response capabilities. Moreover, gesture-based communication enables emergency responders to effectively signal for help or convey critical information in situations where verbal communication may be

restricted or impractical. These applications leverage advanced vision-based recognition systems to accurately interpret gestures, ensuring swift and informed responses in critical situations. [34].

(Uses vision-based recognition to analyze and interpret gestures in real-time surveillance footage.)

4.2 Sensor - Based Gesture Recognition

Sensor-based approaches integrate technologies like surface electromyography (sEMG), ultrasound (US), flex sensors, inertial measurement units (IMUs), and cameras to capture gesture signals. Signal processing techniques, feature extraction methods, and fusion algorithms are utilized to interpret gestures. Applications range from prosthetic control and rehabilitation to sign language recognition, with ongoing efforts to enhance sensor wearability, real-time processing capabilities, and data privacy in wearable devices [33].

4.2.1 Technology Overview

The technologies employed in the studies included Surface Electromyography (sEMG) Sensors, which detect electrical signals produced by muscle contractions[33]. Ultrasound (US) Probes were used to emit ultrasonic pulses for detecting muscle movements and shapes [55]. Inertial Measurement Units (IMUs) were utilized to capture orientation and motion data during the experiments.

4.2.2 Algorithms

The studies utilized a variety of advanced algorithms to enhance the analysis and recognition of gestures. Signal processing techniques, including filtering and feature extraction methods such as time-domain and frequency-domain analyses, along with classification algorithms like Support Vector Machines (SVM) and Artificial Neural Networks (ANN), were extensively applied to process sEMG signals effectively[55]. Additionally, fusion algorithms were implemented to integrate data from multiple sensors, thereby improving the overall accuracy and robustness of gesture recognition systems. Furthermore, the application of Grasshopper Optimization proved instrumental in refining segmentation and classification tasks, particularly beneficial in the intricate realm of 3D hand gesture recognition [56]. These algorithmic approaches collectively contribute to advancing the capabilities of gesture recognition technology across various practical domains.

4.2.3 Applications

Healthcare and Rehabilitation:

Gesture recognition technology finds valuable applications in healthcare and rehabilitation settings. In assistive technologies for individuals with disabilities, it enables intuitive control of prosthetic limbs, wheelchairs, and other assistive devices through hand gestures, as discussed in [33]. Additionally, in physical therapy rehabilitation tools, patients can engage in exercises and interact with virtual environments using gestures, which aids in their recovery process. These applications integrate sensor-based recognition methodologies such as surface electromyography (sEMG) and ultrasound (US) for detecting muscle movements and shapes, as detailed in [55]. By leveraging these technologies, healthcare professionals can offer more personalized and effective rehabilitation programs, enhancing patient mobility and overall quality of life.

4.3 Hybrid Gesture Recognition

Hybrid approaches harness the power of neural networks, facilitated by advanced hardware such as GPUs and specialized computing platforms. Models like Mask-RCNN with [56]ResNet backbones, Grasshopper Optimization, and various CNN architectures are employed for 3D and RGB hand gestures segmentation and classification. These systems enable real-time human and gesture recognition for UAV rescue operations, showcasing high accuracy and robustness in complex environments [34].

4.3.1 Technology Overview

Deep Neural Networks (DNNs) are employed extensively to learn intricate patterns within data sets, enabling sophisticated analysis and prediction capabilities. Convolutional Neural Networks (CNNs) prove highly effective in tasks involving image recognition and processing, leveraging their ability to extract meaningful features from visual data[56]. Recurrent Neural Networks (RNNs) are instrumental in handling sequential data, making them well-suited for tasks that involve time-series analysis, natural language processing, and other sequential data processing applications. These neural network architectures collectively enhance the capabilities of gesture recognition systems, enabling robust and accurate performance across a wide range of applications and domains

4.3.2 Algorithms

Support Vector Machines (SVM) are utilized for effective gesture classification and pattern recognition tasks. Feed Forward Back Propagation Artificial Neural Networks (ANN) are employed for their ability to classify gestures accurately based on learned features. Graph Convolutional Neural Networks (Graph CNN) excel in 3D hand pose estimation tasks, leveraging graph structures to model relationships between keypoints. Additionally, the Self-Organizing Hand Network (SO—Hand Net) employs semi-supervised learning techniques to enhance 3D hand pose estimation accuracy by leveraging unlabeled data. These models and architectures play crucial roles in advancing gesture recognition technology, enabling applications across various domains including healthcare, robotics, and virtual reality.

4.3.3 Applications

1. Smart Home Automation:

Gesture-based control of smart home devices, including lights, thermostats, and entertainment systems, allowing users to adjust settings and activate functions with simple hand gestures [33], [55].

(Employs deep learning algorithms for gesture recognition and classification, enabling seamless integration with smart home automation systems.)

2.Education and Training:

Gesture recognition technology enhances interactive learning experiences in classrooms and training environments by allowing students to engage with educational content and simulations using hand gestures, thereby improving understanding and retention. Instructors benefit from gesture-based feedback and assessment tools that enable them to evaluate student performance and provide personalized guidance. These applications utilize deep learning techniques to analyze and interpret gestures effectively within educational contexts, facilitating more immersive and effective learning environments. This technology not only enhances engagement but also supports tailored educational approaches that cater to individual learning styles and needs.

3.Remote Control and Robotics:

Gesture recognition technology is instrumental in enabling remote control of unmanned aerial vehicles (UAVs) and robots for diverse tasks such as search and rescue operations, environmental monitoring, and inspection of hazardous environments[34]. Additionally, in collaborative robotics applications within manufacturing and assembly settings, human operators utilize hand gestures to control robotic arms and manipulate objects with precision and coordination. These applications leverage deep learning algorithms for robust gesture recognition and seamless control of robotic systems, enhancing operational efficiency and flexibility in remote and collaborative environments. By integrating gesture recognition technology, these industries benefit from improved safety, increased productivity, and expanded capabilities in complex operational scenarios.

4.4 Future Directions

- Integration of Multi-Modal Data: Future research could explore methods for integrating multi-modal data, combining vision-based and sensor-based approaches to enhance robustness and accuracy in gesture recognition systems.
- Real-Time Performance: Developing efficient algorithms and hardware solutions for real-time processing is crucial to enable seamless interaction in dynamic environments, such as rescue operations and human-computer interfaces [33].
- Wearable Gesture Recognition: Further advancements in wearable sensor technology could lead to the development of compact and unobtrusive devices for gesture recognition, facilitating applications in healthcare, rehabilitation, and consumer electronics [55].
- Privacy and Security: Addressing concerns related to data privacy and security is paramount, necessitating the implementation of encryption and authentication mechanisms to protect user information in gesture recognition systems.
- Adaptive Learning: Incorporating adaptive learning techniques, including online learning and reinforcement learning, could enhance system adaptability and responsiveness to user preferences and environmental changes.

- Human-Centric Design: Designing gesture recognition systems with a focus on user experience and human factors is essential to ensure usability and acceptance across diverse user groups, including individuals with disabilities [55].

In conclusion, ongoing advancements in technologies and algorithms for hand gesture recognition hold promise for revolutionizing human-machine interaction. Future research endeavors should focus on addressing existing challenges, enhancing system performance, and promoting inclusive and accessible interaction paradigms for users worldwide.

4.5 Conclusion

The reviewed articles collectively demonstrate the diverse methodologies and advanced technologies employed in hand gesture recognition systems. Vision-based techniques, including appearance-based, motion-based, skeleton-based, depth-based, and 3D model-based recognition, leverage image processing, machine learning, and depth sensing to enable natural interaction with devices. Sensor-based approaches, such as surface electromyography (sEMG) and ultrasound (US) technologies, offer promising alternatives, particularly for deeper muscle detection and enhanced accuracy. Deep learning methods, highlighted in several papers, exhibit significant potential, with convolutional neural networks (CNNs) and recurrent neural networks (RNNs) enabling complex pattern recognition and pose estimation tasks. These approaches, coupled with advancements in sensor technology and algorithm development, pave the way for more intuitive human-machine interactions across diverse application domains.

These technologies and algorithms play essential roles in various aspects of hand gesture recognition, from data acquisition and preprocessing to feature extraction, classification, and pose estimation. They enable systems to interpret and respond to human gestures accurately, facilitating intuitive interactions with devices and systems in different application domains.

CHAPTER 5: DEMONSTRATION OF VARIOUS GESTURE RECOGNITION PLATFORMS

5.1 Mediapipe

Mediapipe (mediapipe-studio.webapps.google.com) is a software framework developed by Google to facilitate the development of applications that can analyze and understand visual and audio data. It offers a collection of tools and pre-built components that enable tasks such as hand tracking, face detection, gesture recognition, and more. These tools streamline the development process, allowing developers to create sophisticated applications with ease. Mediapipe is particularly useful for research projects, as it provides a flexible and efficient platform for experimenting with machine learning algorithms and perceptual computing tasks. Its modular architecture and support for various programming languages make it suitable for a wide range of projects, from simple prototypes to complex applications. By leveraging the capabilities of Mediapipe, researchers can focus on their core objectives without getting bogged down by the intricacies of building and optimizing perception-based systems from scratch.

3.3.3 Technical specifications and capabilities of Mediapipe

Typically, image or video input data is retrieved in separate streams and analyzed using neural networks like TensorFlow, PyTorch, CNTK, or MXNet. These models follow a simple and deterministic approach: one input yields one output, ensuring efficient processing. Conversely, MediaPipe operates at a higher semantic level, allowing for more intricate and dynamic behavior. For instance, a single input in MediaPipe can generate zero, one, or multiple outputs, a capability beyond the scope of traditional neural networks. In the realm of video processing and AI perception, streaming processing is preferred over batching methods. While OpenCV 4.0 introduced the Graph API for sequencing OpenCV image processing operations as a graph, MediaPipe offers versatility by supporting operations on various data types and providing native streaming support for time-series data. This feature makes MediaPipe particularly well-suited for analyzing audio and sensor data.

MediaPipe Framework consists of three main elements: a framework for inference from sensory data (audio or video), a set of tools for performance evaluation, and re-usable components for inference and processing (calculators). The fundamental components of MediaPipe include Packets, Graphs, Nodes, and Streams. These elements define a data-flow graph where packets flow across nodes collated by timestamps within time-series streams. The pipeline can be refined incrementally by inserting or replacing calculators anywhere in the graph, and custom calculators can be defined without specialized expertise in multithreaded programming.

MediaPipe supports GPU computing and rendering nodes, allowing for the combination of multiple GPU nodes with CPU-based nodes. Different GPU APIs can be utilized on mobile platforms, and individual nodes can be written using different APIs to take advantage of platform-specific features. The MediaPipe tracer module records timing events across the graph, aiding in performance tuning by reporting average and extreme latencies. The recorded timing data also helps diagnose various problems such as unexpected real-time delays and memory accumulation.

The MediaPipe visualizer tool facilitates understanding the topology and behavior of pipelines through timeline and graph views. The timeline view displays precise timings of data movement through threads and calculators, while the graph view visualizes the topology of the graph and the state of each calculator. In summary, Google MediaPipe offers a versatile framework for creating customizable machine learning solutions for live and streaming media, making it suitable for a wide range of computer vision applications [57].

3.3.4 *Gesture based Sign Language Recognition system using Mediapipe*

People with hearing impairments often encounter significant difficulties in communicating with those who can hear. Sign language serves as one of the most effective means of communication for them. However, understanding sign language poses a challenge for individuals who are not familiar with it. This communication barrier can negatively affect the social and emotional well-being of people with hearing disabilities, hindering their full participation in society.

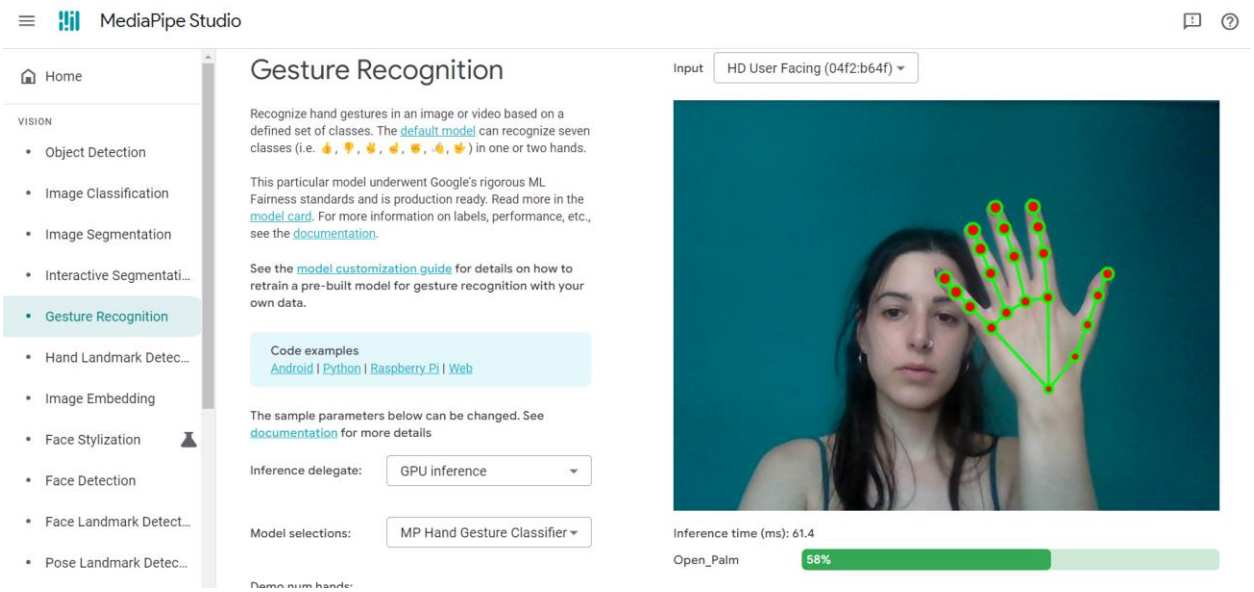


Image 15: MediaPipe Studio (Source: mediapipe-studio.webapps.google.com)

The proposed [58] Sign Language Recognition system presents an innovative solution to bridge the communication gap between individuals with hearing impairments and those who can hear. This system demonstrates high accuracy in recognizing sign language, achieving a classification accuracy of 98.975% with an SVM model. Additionally, the integration of Google's MediaPipe palm detector method has made the system accessible to users without requiring special hardware, representing a significant advantage.

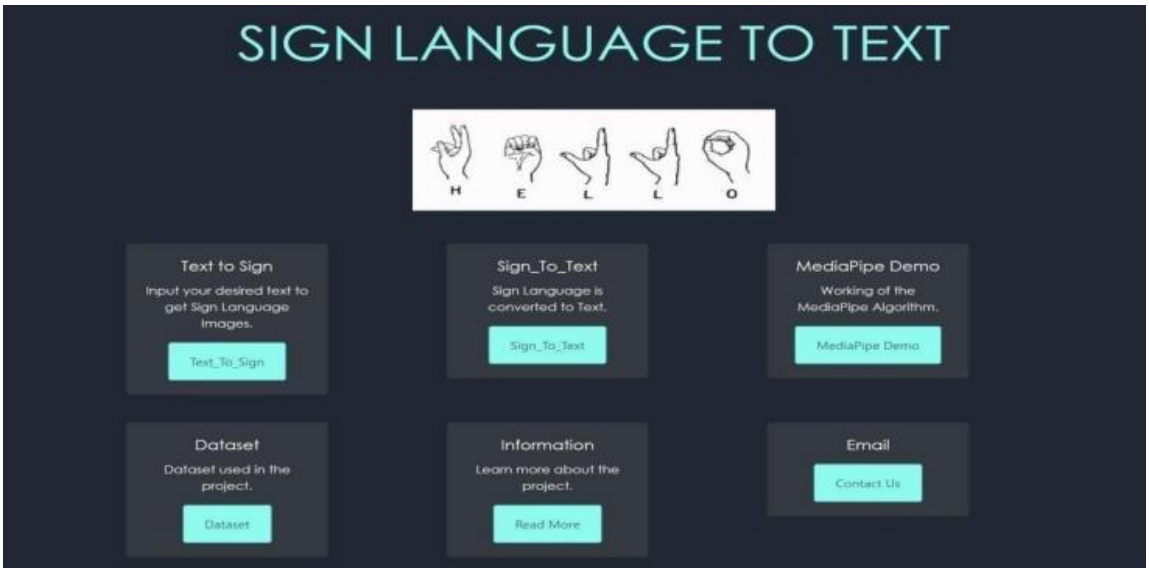


Image 16: *Homepage of Sign Language app. (Source: assets.researchsquare.com)*

The potential practical applications of the proposed method are substantial, offering the possibility of enhancing the quality of life for individuals with hearing impairments by facilitating communication with the wider community. Future endeavors will focus on expanding the current system to incorporate more gestures and developing a comprehensive and reliable system for mobile platforms.

5.2 Leap Motion

Leap Motion (https://www.ultraleap.com/) is a hand-tracking platform renowned for its ability to enable natural and intuitive interaction with computers and virtual environments. Comprising a compact hardware device equipped with advanced infrared cameras and sensors, Leap Motion accurately tracks hand and finger movements in three dimensions. Developers leverage the Leap Motion SDK to integrate this technology into applications, empowering users to control software through gestures like swipes, pinches, and grabs. Widely utilized in virtual reality (VR) and augmented reality (AR) experiences, as well as medical simulations and educational software, Leap Motion offers precise hand tracking that fosters immersive and innovative interactions, shaping the landscape of human-computer interaction.

5.2.1 Technical specifications and capabilities of Leap Motion

Until now, most interactions with computer programs have involved an intermediary step, such as using a mouse or keyboard, between the human hand and the digital environment. The Leap Motion Controller represents a significant advancement in bridging this gap by enabling humans to manipulate computer programs in a way that closely resembles how they interact with real-world objects. The functionality of the Leap Motion Controller is restricted to programs that are explicitly designed to be compatible with it. This limitation confines its capabilities to the features and functions integrated into these dedicated programs. Additionally, although many of the compatible software applications perform effectively, there is a tendency for inconsistency in their operation [59].

The creative potential, on the other hand, of the Leap Motion Controller is immense due to developers being able to design their own software tailored for it. Leap Motion has empowered creators by providing them with the tools to experiment with and customize the technology.

Consequently, this has spurred the development of novel software and innovative applications for this technology.

Leap Motion primarily uses C++ for its core implementation. However, the Leap Motion SDK supports multiple programming languages, allowing developers to integrate and use Leap Motion functionalities in various environments. These languages include:

- C++: The core SDK is implemented in C++ and provides comprehensive features and high performance for developing applications.
- JavaScript: The LeapJS library allows web developers to create applications using Leap Motion in the browser.
- C#: For Windows applications, especially those built with Unity, C# is supported through the Leap Motion Unity SDK.
- Python: The SDK provides bindings for Python, enabling rapid prototyping and development in Python environments.
- Java: Java bindings are available, allowing the use of Leap Motion in Java-based applications.

By supporting these languages, Leap Motion ensures that developers from various backgrounds can leverage its technology in their preferred development environment.

5.2.2 Examples of applications developed using Leap Motion

The Leap Motion Controller may not yet be widespread, but it's undeniably fascinating. Its hardware is impressive, but its software capabilities are where things really get interesting. It has the potential to introduce intuitive 3D and gestural controls to various visual tasks on our computers. While this potential is still being realized, the Leap Motion Controller offers a peek into the future of computer interfaces. Users can explore the Airspace Store for apps that showcase its capabilities, though some are more like technical demonstrations and toys, while others offer deeper functionalities.

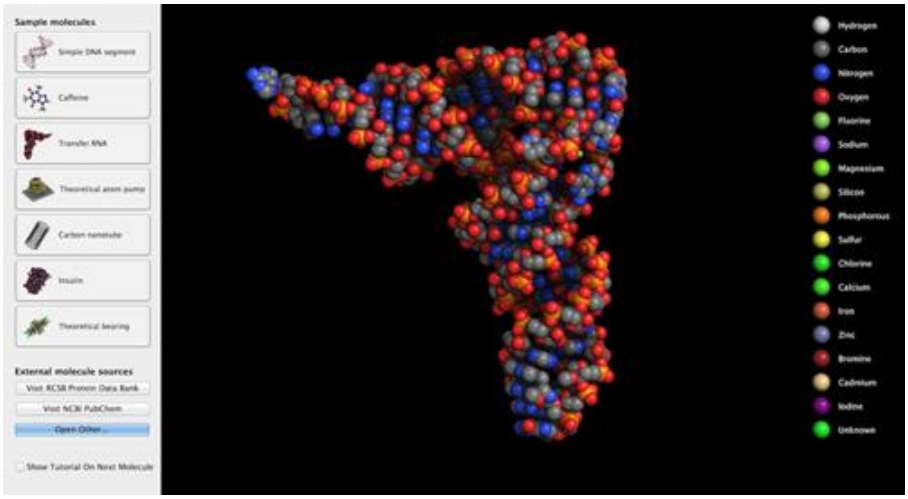
Education is a particularly promising area of app development, offering immersive experiences in three dimensions. Gaming apps also demonstrate the unique aspects of the controller while providing entertainment. Additionally, some apps offer genuinely useful capabilities, from

creating digital art to navigating desktops and browsers. The Leap Motion Controller offers a fresh way to interact with PCs, alongside the traditional keyboard and mouse.

While there's still room for developers to explore, the initial batch of apps already showcases the diverse potential of this technology.

Below are some images showcasing the area of Education using Leap Motion:

1. Molecules



Molecules is an application exclusive to Mac devices, designed for viewing and interacting with 3D representations of molecules. Users can navigate around the molecule to gain a better grasp of its shape and zoom in and out to explore its structure and chemical bonds hands-on.

Image 17: *Leap Motion Molecules (Source: www.pcmag.com)*

2. Cyber Science – Motion

Cyber Science introduces Motion 3D, a feature that brings anatomy knowledge to life in three dimensions. With Motion 3D, users can explore, dissect, and reassemble a human skull while learning the names and locations of all 28 bones.

Image 18: *Cyber Science – Motion (Source: [58])*

Users can opt for a relaxed mode, where they can freely pan, dissect, and

reassemble the skull at their own pace. Alternatively, they can choose from 13 challenge modes that test their skills against the clock as they reassemble the intricate jigsaw puzzle of the human skull [60].

3. Exoplanet

Exoplanet, available exclusively for Mac, offers an immersive experience by placing the Milky Way at your fingertips. Users can delve into our solar system or navigate through the extensive catalog of known planets, continuously updated with new discoveries.

With Exoplanet, users can soar through space and explore the universe on multiple levels, all made possible through interactive navigation with the Leap Motion Controller.

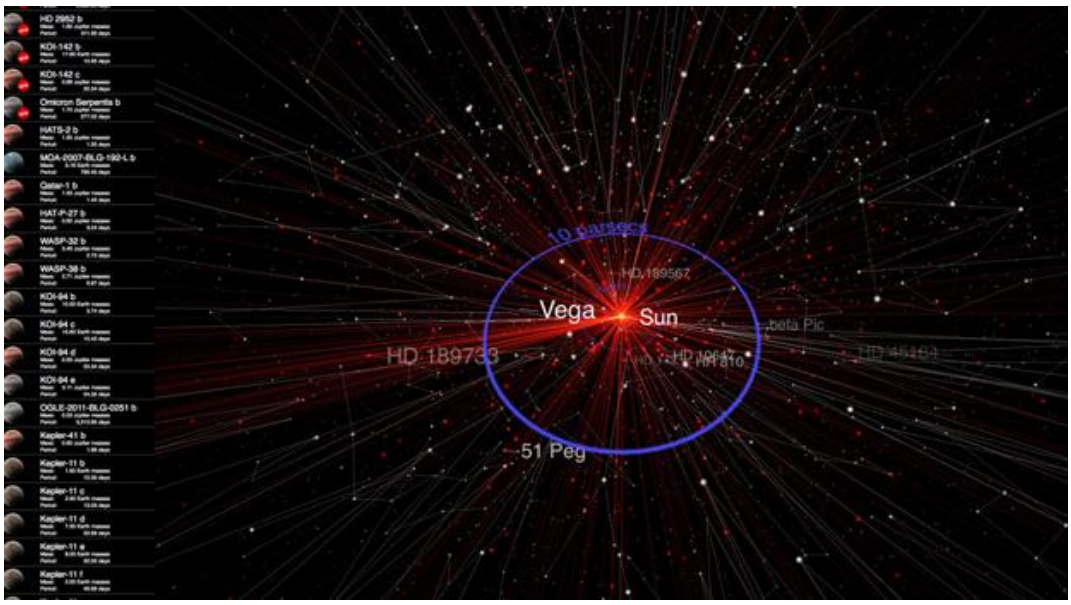


***Image 19**: Exoplanet (Source:[58])*

5.3 Intel RealSense

Intel RealSense Technology, previously known as Intel Perceptual Computing, comprises a suite of depth and tracking technologies aimed at providing machines and devices with depth perception capabilities. Owned by Intel, these technologies are utilized in various applications such as autonomous drones, robots, augmented reality (AR) and virtual reality (VR) systems, and smart home devices, among others.

The RealSense product line consists of Vision Processors, Depth and Tracking Modules, and Depth Cameras. These components are supported by an open-source, cross-platform Software Development Kit (SDK), which aims to simplify the integration of RealSense cameras for third-party software developers, system integrators, Original Design Manufacturers (ODMs), and Original Equipment Manufacturers (OEMs).

5.3.1 Technical specifications and capabilities of Intel RealSense

A. Intel RealSense Depth Camera D415, D435 and D455

The depth technology utilized by the system is active infrared (IR) stereo [61], [62], [63], [64] [65], suitable for both indoor and outdoor environments. It comprises an IR projector and left/right cameras, available in standard and wide types, respectively. The shutter type employed is rolling for the standard camera and global for the wide camera. The image sensor modules consist of the OV2740 (OV02740-H34A-Z) for the standard camera and the OV9782 (OV09782-GA4A) for the wide camera, employing PureCel HDR and OmniPixel3-GS technologies, respectively. The image sensor sizes differ, with the standard camera featuring a 1/6 inch sensor and the wide camera featuring a 1/4 inch sensor. Furthermore, the vision processor board utilized is the RealSense Vision Processor D4. The depth sensor modules include the RealSense Module D415, RealSense Module D430 + RGB Camera, and RealSense Module D450, each with varying depth field of view, depth resolution, and framerate specifications. The RGB camera resolution, framerate, and aspect ratio also vary accordingly. The device dimensions and mounting mechanisms are provided for ease of installation and integration, with connectivity supported through a USB Type-C 3.1 Gen 1 connector.

B. Intel RealSense Vision Processor D4 Series

The RealSense Vision Processor[66] D4 and RealSense Vision Processor D4M both utilize stereo depth technology and come in ASIC BGA form factors. However, they differ in package size, with the D4 measuring 6.4mm x 6.4mm x 1mm and the D4M measuring 4.7mm x 3.8mm x 0.55mm. Both processors operate on a 28nm process technology.

In terms of depth capabilities, the D4 has a depth max throughput of 36.6 MP/sec (848×480@90fps) and supports depth stream output resolutions of up to 1280×720 at up to 90fps. On the other hand, the D4M supports depth stream output resolutions of up to 720×720.

For RGB sensor capabilities, the D4 supports a maximum resolution of 1920×1080 at up to 60fps, while the D4M supports a maximum resolution of 720×720 at up to 30 fps.

Both processors feature IR projector controls, but they differ in host interface and multi-camera support. The D4 interfaces via USB 3.0 and supports multi-camera setups of up to 5 cameras, while the D4M interfaces via 2x MIPI and supports multi-camera setups at up to 30fps. Additionally, they have different I/O configurations, with the D4 offering 5x MIPI CSI-2, 5x I2C, 1x SPI, GPIO, and Timer, and the D4M offering 2x MIPI, 1x I2C, 1x SPI, GPIO, and Timer.

C. Intel Stereo DepthModule SKUs

The RealSense D400 [61] series, consisting of models D410, D415, D420, and D430, feature various depth technologies and image sensor technologies tailored for specific applications.

The D410 and D415 utilize Active IR Stereo depth technology, while the D420 and D430 employ Passive IR Stereo. Additionally, the D400 series employs Rolling Shutter image sensor technology in the D410, D415, and D420 models, while the D420 and D430 feature Global Shutter.

Regarding depth field of view (FOV), all models exhibit a FOV of approximately 63.4 degrees horizontally by 40.4 degrees vertically for HD 16:9 resolutions, except for the D420 and D430, which offer a wider FOV of 85.2 degrees horizontally by 58 degrees vertically.

The depth resolution across the series is consistent, with all models supporting resolutions of up to 1280x720. Similarly, the depth frame rate is uniform, with all models supporting frame rates of up to 90fps.

In terms of range, all models support a range from 0.16 to over 10 meters, ensuring flexibility in various depth sensing applications. However, the D420 and D430 offer a slightly shorter minimum range, starting from 0.11 meters.

5.3.2 Robotic automation for smart agriculture and renewable energy using Intel RealSense

Autonomous vehicles designed for applications in smart agriculture and renewable energy traditionally rely on Global Positioning System (GPS) data for navigation. However, GPS navigation proves ineffective in structured environments like orchards and solar farms, where signals can be occluded and noisy.

Land Care Robots have innovatively overcome this challenge by employing Intel RealSense depth cameras in tandem with programmable Raspberry Pi 4 RP2040 microcontrollers for autonomous navigation. Unlike traditional methods requiring April tags, QR codes, or other localization symbols, these robots perceive their surroundings in real-time using RealSense depth cameras. This enables them to navigate autonomously and adaptively, without the need for external markers or predefined maps.

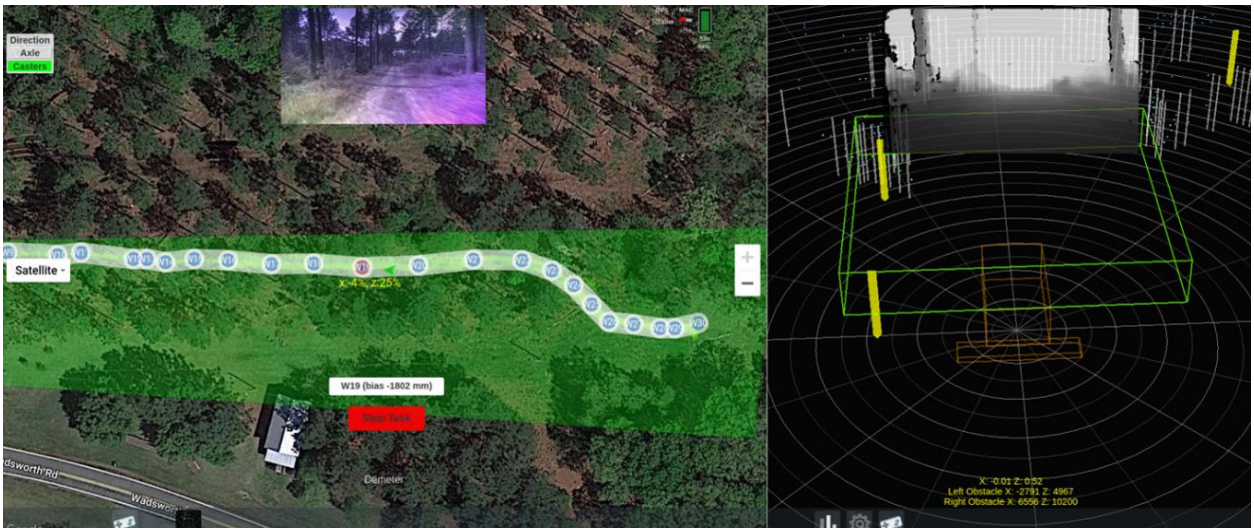


Image 20: *Robotic automation for smart agriculture and renewable energy using Intel RealSense (Source: /www.intelrealsense.com)*

The Land Care Robots (LCR) sample depth images at a rate of 15 frames per second (FPS). These depth images undergo several layers of heavy filtering to eliminate artifacts and transform them into data representations suitable for efficient CPU manipulation. The filtering process includes semantic filtering, 2D filtering, and 1D filtering. Utilizing Intel RealSense technology, much of the depth data processing is performed within the camera itself, thereby relieving the robot's CPU of a significant computational burden.

The Directed Machine's sensor suite is anchored by two Intel RealSense Depth Camera D455s, each equipped with an inertial measurement unit (IMU) for enhanced depth awareness. Each camera produces two data streams: RGB data and depth perception data. In addition to facilitating navigation, the Intel RealSense depth cameras provide the LCRs with dynamic obstacle avoidance capabilities. Thanks to their compact form factor, the cameras can be easily and unobtrusively mounted on the LCR's solar panel [67].

.

CHAPTER 6: IMPLEMENTATION ON A UGV

6.1 Introduction to Unmanned Ground Vehicles (UGVs) and their applications

In its most general sense[68], an unmanned ground vehicle (UGV) refers to any mechanized apparatus designed to traverse the ground surface while carrying or transporting items without accommodating a human operator. A discussion on the wide array of potential UGV systems requires a systematic approach. One such method involves organizing UGV systems based on various characteristics, which may include:

- The purpose behind the development effort, often revolving around fulfilling specific mission requirements.
- The rationale for selecting a UGV solution for the application, such as operating in hazardous environments, meeting strength or endurance needs, or adhering to size constraints.
- Identifying primary technological challenges, such as functionality, performance, or cost concerns, associated with the application.
- The intended operational environment, whether indoor spaces, outdoor roads, rugged terrain, or even underwater locations.
- The method of locomotion employed by the vehicle, such as wheels, tracks, or legs.
- The approach to controlling and navigating the vehicle, encompassing various techniques and technologies utilized for path determination [69].

Human-robot interaction (HRI) in unmanned ground vehicles (UGVs) is crucial for enhancing their usability and effectiveness across various applications. Effective HRI ensures that UGVs can provide real-time feedback and augmented situational awareness, allowing human operators to make informed decisions and respond adaptively to dynamic environments. It fosters operational efficiency by enabling intuitive control systems and dynamic task reallocation, reducing the cognitive load on operators and building trust in the technology. Additionally, HRI enhances safety by allowing human oversight to detect and correct errors, and it promotes better team integration, facilitating seamless collaboration between humans and robots. Ultimately, HRI in UGVs leads to

more successful missions by combining the strengths of human intuition with the precision and capabilities of robotic systems.

6.2 Introducing Azyx IM-1

The Azyx IM-1 («Άζυξ») represents a collaboration between the Laboratory of Machines, Intelligent and Distributed Systems, Hellenic Military Academy with Infili Technologies SA to create a sophisticated Smart Unmanned Ground Vehicle.

Infili is a research-intensive SME that utilizes a unique combination of high-end technologies as an outcome of many years of R&D experience. The company designs solutions for vertical industries with diverse and very demanding requirements regarding the exploitation of their information and knowledge repositories.

6.2.1 Technical Specifications of the Azyx platform

The Azyx platform is a land-based unmanned ground vehicle (UGV) engineered for missions demanding substantial transport capacity. It boasts versatility, accommodating various payloads to suit specific mission requirements. A key feature is its utilization of open technologies in software and electronics, adhering to tested industrial standards. This ensures adaptability and flexibility, allowing seamless integration with different payloads and mission needs. The advanced tailor-made motor controller, powered by an STM32 microcontroller with a 64-bit high-performance processor, supports both RS232 and RS485 interfaces and offers high precision control for optimum motor performance. Additional features include support for Quadrature Encoders, thermal protection, current sensing, and user-selectable peak current, among others. Notably, the Azyx platform is domestically designed and manufactured in Greece, with materials primarily sourced from Greek, European, or American suppliers to ensure a secure supply chain. Furthermore, its payload flexibility enables integration of various options, including armor, heavy infantry weapons, communication or electronic warfare cages, fire suppression systems, and surveillance systems, enhancing its adaptability to diverse operational scenarios.

6.2.2 The Azyx platform Chassis/Body

The chassis/body of the Azyx platform is constructed with a robust steel frame, utilizing steel springs that undergo deformation through pressing. Following this process, they are treated with manganese phosphating and coated with weather-resistant paint, ensuring durability in extreme weather conditions.

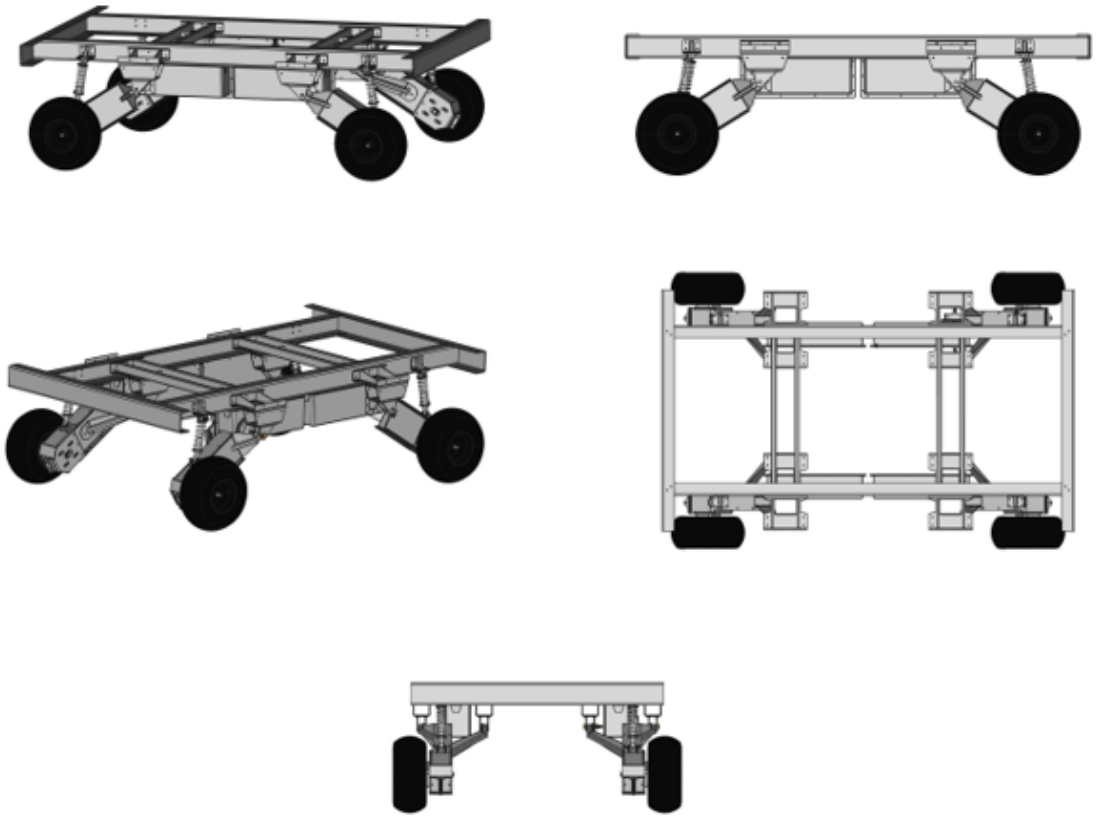


Image 21: *The Azyx platform body*

The platform boasts a remarkable load-bearing capacity, capable of supporting payloads exceeding 500 kg. Additionally, provisions for future reinforcement of the steel frame enable the possibility of significantly enhancing its strength and transport capacity to over 1 ton, without incurring substantial additional construction costs. Emphasizing off-road mobility, the design prioritizes resilience and capabilities tailored for efficient movement in challenging terrain conditions.

6.2.3 The Azyx platform Powerplant

The Azyx platform is equipped with a power system comprising lithium batteries, currently operating at a nominal voltage of 48 VDC. Offering a balance between mobility and endurance, it provides an average autonomy of four (4) hours in motion and over twenty (>20) hours in standby mode, ensuring continuous functionality of its electronic systems. Moreover, the platform features an expandable configuration, allowing for the installation of additional battery arrays or alternative power sources like a diesel generator or fuel cell, thereby extending its autonomy to meet varying operational demands.

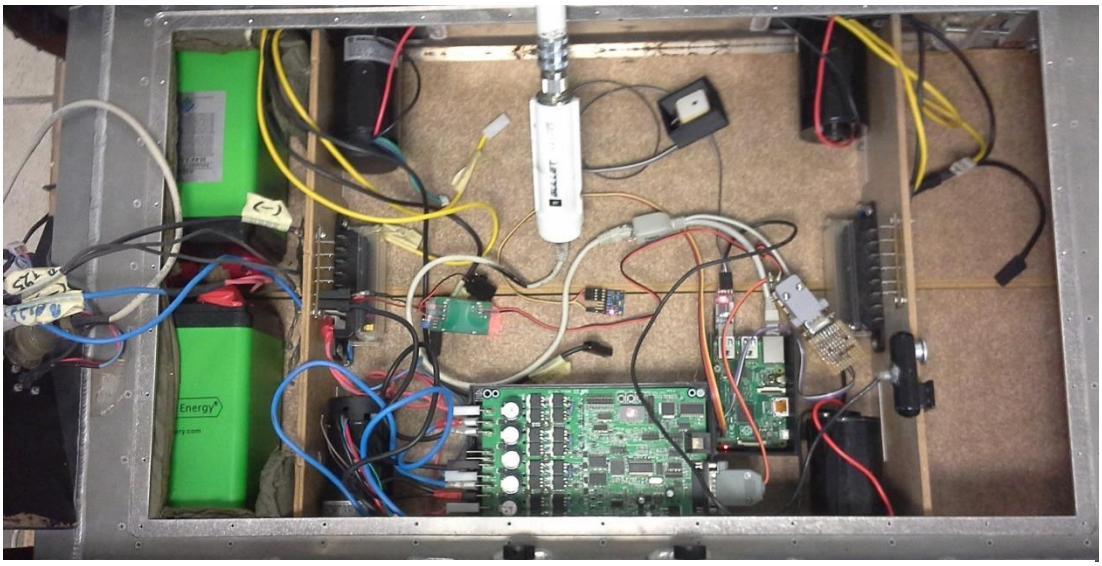


Image 22: *The Azyx platform system*

6.2.4 The Azyx platform Transmission

The transmission system of the Azyx platform features individual, independent electric motors for each wheel, as seen in image 23 below, ensuring optimal performance and control. These motors are Servomotor-Type DC Permanent Magnet Electric Motors, engineered to deliver high torque and designed for variable operation. The final stage of transmission to the wheel involves angular gear reduction, utilizing gears and chains. Importantly, the gear ratio of the chain can be adjusted before each mission to prioritize either torque or speed, enhancing the platform's adaptability to diverse terrain and operational requirements. Additionally, the Azyx platform is equipped with four (4) in-house made motor controllers, specially designed and sourced to efficiently manage each of the vehicle's wheels, ensuring precise and coordinated movement.

Image 23: *The Azyx platform Propulsion*

6.2.5 The Azyx platform Propulsion

The propulsion system of the Azyx platform comprises four rubber tires, as seen in image 23 above, specifically ATV-type, designed without inner tubes for improved durability. Each wheel features independent drive and suspension systems, contributing to enhanced maneuverability and adaptability across various terrains. Directional control is facilitated through differential steering, enabling precise navigation by adjusting the rotation of individual wheels. Moreover, the platform offers wheel customization, allowing users to select the size and type of wheels according to the specific requirements of each mission, with a maximum total diameter of 18 inches, ensuring optimal performance in diverse operational scenarios.

6.2.6 The Azyx platform Power grid and Power Distribution

The Azyx platform relies on a primary power supply directly sourced from the batteries at 48 VDC. Additionally, it incorporates an auxiliary power source to activate in case of primary source failure or interruption, ensuring the continuous operation of electronic systems (excluding electric motors) for safety purposes. All electronic subsystems are powered by circuits incorporating the backup power source to guarantee uninterrupted functionality. The total available consumption for the platform is rated at 16A@48 VDC, providing ample power for its various systems and operations.

6.2.7 The Azyx platform Data Bus

The Azyx platform employs Gigabit Ethernet communication for all inter-subsystem communication, except in cases where it is infeasible or impractical. Communication between the

electric motor controllers and the control computing unit utilizes a Multi-Unit TTL Serial bus, a protocol specifically tailored to the controllers. To facilitate communication, a 5-port Gigabit switch is utilized, with at least 2 ports consistently in use. However, expanding the available ports necessitates a redesign of the communication network architecture due to the current router firmware limitation of 5 total ports.

6.2.8 The Azyx platform Electronics Grid, based on a RPi Computing Unit

The Azyx platform's electronics grid is centered around an RPi computing unit, which handles fundamental tasks such as controlling the electric motor controllers. Acting as a communication node is the Ubiquity Rocket M2 router, enhanced with suitable third-party firmware. Additionally, a real-time microcontroller, equipped with specialized firmware, ensures the efficient functioning of the Power Distribution subsystem, contributing to the overall performance and reliability of the platform. Continuing with the Azyx platform's electronics, a set of cameras is integrated for navigation and remote-control purposes. The control station comprises standard hardware and software components, alongside a Gigabit Ethernet switch and a Ubiquity Rocket M2 router serving as a communication node.

***Image 24**: RPi Computing Unit*

***Image 25:** The Axyx platforms set of cameras.*

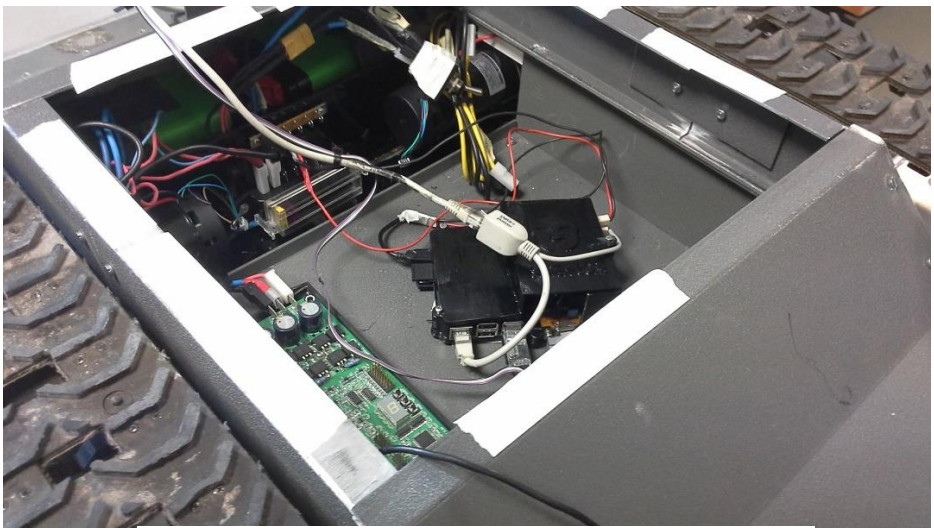


***Image 26**: The Azyx platform Electronics Grid*

Looking ahead, plans include incorporating an nVidia Jetson computing unit tailored for computer vision applications, further enhancing the platform's capabilities in visual perception and processing tasks.

6.2.9 The Azyx platform Network Architecture

The Azyx platform's network architecture is configured as a resilient and adaptable mesh, employing the Mobile Ad-hoc Network (MANET) model. Routing within this network is facilitated by the OLSR protocol, ensuring scalability as needed. Real-time data transmission is achieved with minimal latency (<150 msec) as long as the route comprises four or fewer nodes. Moving forward, future enhancements will include integrating the BATMAN protocol and implementing COMSEC functions for encryption, depending on available wireless options. Additionally, the platform prioritizes IP-based networking and communication methods as a fundamental requirement to ensure compatibility and interoperability across its systems and components.

6.2 Integration of Gesture Recognition Technology

Integrating gesture recognition technology on unmanned ground vehicles (UGVs) using MediaPipe Studio involves leveraging MediaPipe's pre-built models for hand and body tracking to en-

able UGVs to understand and respond to human gestures. The process includes setting up MediaPipe Studio to capture gesture data through a camera, mapping these gestures to specific commands, and establishing a communication protocol between the device running MediaPipe and the UGV. The UGV's control software is then configured to execute commands based on the recognized gestures. This integration enhances UGV usability by providing intuitive, hands-free control, improving operational efficiency, and offering an accessible alternative to traditional control methods.

6.4 Defining Gesture Recognition Logic

To recognize gestures, one needs to define the logic for detecting specific hand poses. Below are presented the Gestures and Commands used for the purpose of this thesis:

a. Thumbs Up

Command: Move Forward 2 meters,

Reason: Intuitive for signaling approval or 'go'.

***Image 27**: Gesture for "Move forward".*

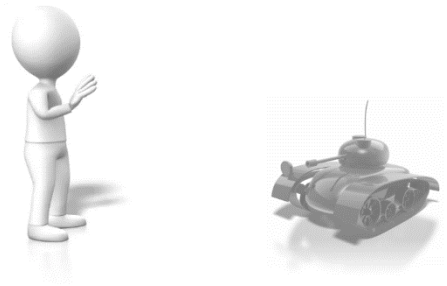
b. Thumbs Down

Command: Move Backward 2 meters,

Reason: Intuitive for signaling disapproval or 'reverse'.

***Image 28**: Gesture for "move backward".*

c. Open Hand (Palm Facing Forward)



Command: Stop

Reason: Clear and easily recognizable for stopping.

Image 29*: Gesture for "Stop".*

d. Pointing Right (Index Finger Pointing Right)

Command: Turn Right

Reason: Directional gestures are intuitive for turning commands.

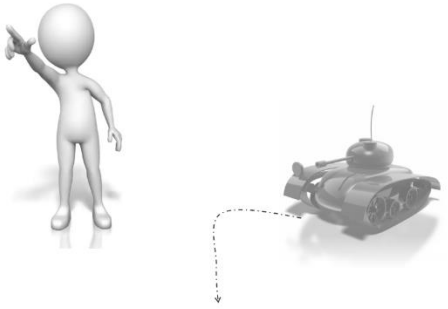
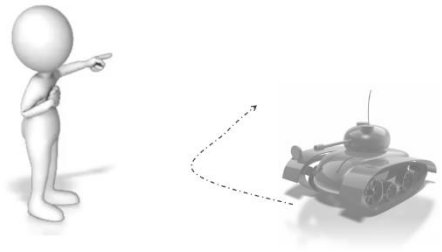


Image 30: *Gesture for "Turn Right".*

e. Pointing Left (Index Finger Pointing Left)



Command: Turn Left

Reason: Same logic as turning right but for the opposite direction.

Image 31*: Gesture for " Turn Left".*

f. Two Fingers Up (Peace Sign)

Command: Increase Speed 5km/h

Reason: Can signify a boost or increase.

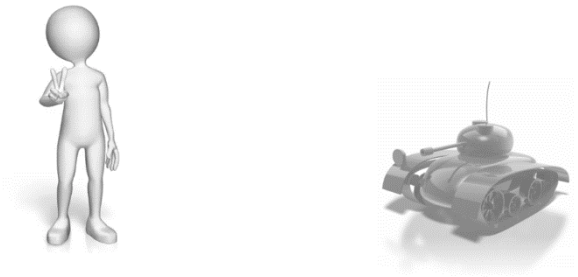


***Image 32**: Gesture for "Increase Speed".*

g. Two Fingers Down (Inverted Peace Sign)

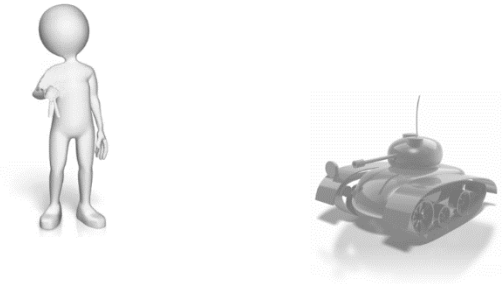


***Image 33**: Gesture for "Decrease Speed".*

Command: Decrease Speed 5km/h

Reason: Can signify a reduction or decrease.

h. Wave (Side to Side)

Command: Switch Mode / Toggle Autonomy

Reason: A waving gesture can be used to toggle between manual and autonomous modes or switch operational modes.

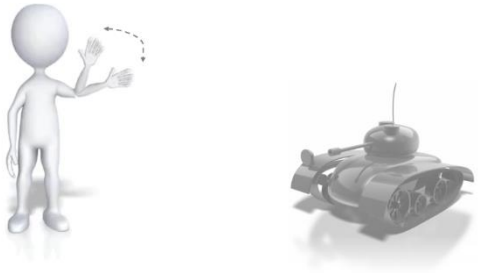


***Image 34**: Gesture for "Switch Mode"*

Assuming you have a UGV with an API or a communication interface, you need to send commands based on recognized gestures. The script captures video input, processes it for hand gestures using MediaPipe, and sends commands to a UGV based on recognized gestures.

CHAPTER 7: DISCUSSION AND FUTURE DIRECTIONS

Gesture recognition technology holds significant potential beyond its application in unmanned ground vehicles (UGVs), extending to various real-life paradigms that can revolutionize human-computer interaction. In the realm of smart homes, gesture recognition can enable intuitive control of home appliances, lighting, and security systems, providing a seamless and hands-free user experience. In healthcare, this technology can assist patients with limited mobility, allowing them to interact with medical devices or summon assistance through simple gestures. Additionally, gesture recognition can enhance gaming and virtual reality environments by providing more immersive and natural interaction methods. In automotive applications, drivers can use gestures to control infotainment systems, navigate interfaces, and manage in-car functions without taking their eyes off the road. The integration of gesture recognition in public spaces, such as interactive kiosks or digital signage, can facilitate touchless interactions, enhancing user convenience and hygiene.

Future directions for gesture recognition technology should focus on improving the accuracy and reliability of gesture detection across diverse environments and user demographics. Enhancing machine learning algorithms to better understand subtle and complex gestures can lead to more nuanced and context-aware interactions. Integrating multimodal input, combining gesture recognition with voice commands and facial expressions, could create more comprehensive and natural user interfaces. Additionally, expanding the use of gesture recognition in assistive technologies for individuals with disabilities can further enhance their independence and quality of life. Research into energy-efficient and miniaturized sensor technology will also be crucial for integrating gesture recognition into wearable devices and other compact applications. As privacy and security remain paramount, developing robust protocols to protect user data in gesture-based systems will be essential. These future directions highlight the need for continued innovation and interdisciplinary collaboration to fully realize the transformative potential of gesture recognition technology.

REFRENCES

[1] N. Saini, “RESEARCH PAPER ON ARTIFICIAL INTELLIGENCE & ITS APPLICATIONS,” *2023 IJRTI*, vol. 8, no. 4, 2023.

[2] D. Leslie, C. Burr, M. Aitken, J. Cowls, M. Katell, and M. Briggs, *Artificial intelligence, human rights, democracy, and the rule of law: a primer. The Council of Europe.*

[3] Y. Xu *et al.*, “Artificial intelligence: A powerful paradigm for scientific research,” vol. 2, no. 4, 2021, doi: https://doi.org/10.1016/j.xinn.2021.100179.

[4] B. J. Copeland and A. Turing, “The Essential Turing: Seminal Writings in Computing, Logic, Philosophy, Artificial Intelligence, and Artificial Life: Plus The Secrets of Enigma.,” 2004.

[5] Q. Liu and Y. Wu, “Supervised Learning,” 2012, doi: 10.1007/978-1-4419-1428-6_451.

[6] T. Neelam, “5 Clustering Methods and Applications.” Accessed: Mar. 25, 2024. [Online]. Available: https://www.analyticssteps.com/blogs/5-clustering-methods-and-applications

[7] C.-Y. Lin, R. C. Weng, and S. S. Keerthi, “Trust Region Newton Method for Logistic Regression.,” vol. 9, pp. 627–650, doi: 10.1145/1390681.1390703.

[8] P. Sheean, B. Bruemmer, P. Gleason, J. Harris, C. Boushey, and L. Horn, “Publishing nutrition research: a review of multivariate techniques--part 1,” *Elsevier Inc.*, vol. 111, no. 1, doi: 10.1016/j.jada.2010.10.010.

[9] Data Base Camp, “What is the Naive Bayes Algorithm?” Accessed: Apr. 23, 2024. [Online]. Available: https://databasecamp.de/en/ml/naive-bayes-algorithm

[10] D. Jurafsky and J. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, vol. 2. Prentice Hall, 2008.

[11] Z. Ghahramani, “Unsupervised Learning,” in *Advanced Lectures on Machine Learning*, Berlin, Germany: Springer Berlin Heidelberg, 2023, pp. 72–112.

[12] Y. Bengio, A. Courville, and P. Vincent, “Representation learning: A review and new perspectives,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, pp. 1798–1828, 2013.

[13] J. Gareth, D. Witten, T. Hastie, and R. Tibshirani, *An introduction to statistical learning: With applications in python*. Springer Nature, 2023.

[14] A. Jain, “Data Clustering: 50 Years Beyond K-Means,” vol. 31, no. 8, pp. 651–666, 2010, doi: 10.1016/j.patrec.2009.09.011.

[15] J. A. Hartigan and A. S. Agoes, “Algorithm AS 136: A K-Means Clustering Algorithm. Journal of the Royal Statistical Society: Series C (Applied Statistics),” vol. 28, pp. 100–108, 1979, doi: doi.org/10.2307/2346830.

[16] J. Lever, M. Krzywinski, and N. Altman, “Points of Significance: Principal component analysis,” *Nature Methods*, vol. 14, no. 7, pp. 641–642, 2017, doi: 10.1038/nmeth.4346.

[17] C. Ma, S. Zhang, A. Wang, Y. Qi, and G. Chen, “Skeleton-Based Dynamic Hand Gesture Recognition Using an Enhanced Network with One-Shot Learning,” vol. 10, no. 11, p. 3680, doi: 10.3390/app10113680.

[18] N. K. Ampah, C. M. Akujuobi, M. N. O. Sadiku, and S. Alam, “An intrusion detection technique based on continuous binary communication channels,” *International J. Security and Networks*, vol. 2, no. 2/3, pp. 174–180, 2011.

[19] H. Dike, Y. Zhou, K. K. Deveerasetty, and Q. Wu, “Unsupervised Learning Based On Artificial Neural Network: A Review,” presented at the 2018 IEEE International Conference on Cyborg and Bionic Systems (CBS), 2018, pp. 322–327. doi: 10.1109/CBS.2018.8612259.

[20] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning with Applications in R*, vol. 103. in Springer Texts in Statistics, vol. 103. Springer.

[21] I. Logunova, “K-Means Clustering in Machine Learning.” Accessed: Mar. 25, 2024. [Online]. Available: https://serokell.io/blog/k-means-clustering-in-machine-learning

[22] T. J. Loftus *et al.*, “Phenotype clustering in health care: A narrative review for clinicians,” vol. 5, doi: 10.3389/frai.2022.842306.

[23] S. Aishwarya, “Reinforcement Learning: The Business Use Case,” 2018.

[24] E. Choi, D. Hewlett, J. Uszkoreit, I. Polosukhin, A. Lacoste, and J. Berant, “Coarse-to-Fine Question Answering for Long Documents,” in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, Vancouver, Canada: Association for Computational Linguistics, 2017, pp. 209–220. doi: 10.18653/v1/P17-1020.

[25] R. Paulus, C. Xiong, and R. Socher, “A DEEP REINFORCED MODEL FOR ABSTRACTIVE SUMMARIZATION”, [Online]. Available: https://arxiv.org/pdf/1705.04304

[26] M. Rouse, “Deep Learning,” *Technopedia*. 2024. Accessed: Feb. 02, 2024. [Online]. Available: https://www.techopedia.com/definition/30325/deep-learning

[27] P. M. Wilamowski, “Neural networks and fuzzy systems for nonlinear applications,” in *11th INES 2007*, Budapest, Hungary, 2007, pp. 13–19.

[28] A. Farooq, “PyTorch vs TensorFlow in 2024: A Comparative Guide of AI Frameworks,” Open CV. Accessed: Mar. 23, 2024. [Online]. Available: https://opencv.org/blog/pytorch-vs-tensorflow/

[29] C. Cortes and V. Vapnik, “Support-vector networks,” vol. 20, pp. 273–297, 1995, doi: doi.org/10.1007/BF00994018.

[30] C. M. Bishop, *Pattern recognition and machine learning*. New York: Springer, 2006.

[31] B. Üstün, “Patterns before recognition: the historical ascendance of an extractive empiricism of forms,” vol. 11, no. 55, 2024, doi: https://doi.org/10.1057/s41599-023-02574-1.

[32] G. R. S. Murthy and R. S. Jadon, “A REVIEW OF VISION BASED HAND GESTURES RECOGNITION,” vol. 2, no. 2, pp. 405–410, 2009.

[33] L. Guo, L. Zongxing, and L. Yao, “Human-Machine Interaction Sensing Technology Based on Hand Gesture Recognition: A Review,” *IEEE Transactions on Human-Machine Systems*, vol. 51, no. 4, pp. 300–309, 2021.

[34] C. Liu and T. Sziranyi, “Real-Time Human Detection and Gesture Recognition for On-Board UAV Rescue,” vol. 21, no. 6, p. 2180, 2021, doi: 10.3390/s21062180.

[35] C. H. Chen and P. G. Ho, “Statistical pattern recognition in remote sensing,” *Pattern Recognition*, vol. 41, no. 9, pp. 2731–2741, 2008, doi: 10.1016/j.patcog.2008.04.013.

[36] T. Pavlidis, *Structural Pattern Recognition*. in Springer Series in Electronics and Photonics. 1977.

[37] J. Liu, J. Sun, and S. Wang, “Pattern Recognition: An overview,” vol. 6, no. 6, 2006.

[38] X. Chen, X. Dong, W. Zeng, and C. Yuan, “UGV Direction Control by Human Arm Gesture Recognition via Deterministic Learning,” in *Proceedings of the 38th Chinese Control Conference*, Guangzhou, China, 2019. doi: 10.23919/ChiCC.2019.8865788.

[39] M. Ouda, A. Al- Naji, and J. Chahl, “Hand Gesture Recognition Based on Computer Vision: A Review of Techniques,” vol. 6, no. 8, p. 73, 2020, doi: https://doi.org/10.3390/jimaging6080073.

[40] A. Choudhury, A. Talukdar, and K. Sarma, “A Review on Vision-Based Hand Gesture Recognition and Applications,” in *Intelligent Applications for Heterogeneous System Modeling and Design*, 1st ed., USA: IGI Global PA, 2015, pp. 261–286.

[41] L. Lamberti and F. Camastra, “Real-Time Hand Gesture Recognition Using a Color Glove,” presented at the Image Analysis and Processing - ICIAP 2011 - 16th International Conference, Ravenna, Italy, 2011, pp. 365–373. doi: 10.1007/978-3-642-24085-0_38.

[42] F. Al Farid *et al.*, “A Structured and Methodological Review on Vision-Based Hand Gesture Recognition System,” *Journal of Imaging*, vol. 8, no. 6, p. 153, 2022, doi: https://doi.org/10.3390/jimaging8060153.

[43] R. Y. Wang and J. Popovi’c, “Real-Time Hand-Tracking with a Color Glove,” presented at the ACM SIGGRAPH 2009 papers, in 3, vol. 28. 2009. doi: 10.1145/1531326.1531369.

[44] D. Kuiaski, H. Viera Neto, G. Benvenutti Borba, and H. Gamba, “A Study of the Effect of Illumination Conditions and Color Spaces on Skin Segmentation,” in *SIBGRAPI*, Rio de Janeiro, Brazil, 2009. doi: 10.1109/SIBGRAPI.2009.47.

[45] V. Nasteski, “An overview of the supervised machine learning methods,” vol. 4, pp. 51–62, 2017, doi: 10.20544/HORIZONS.B.04.1.17.P05.

[46] A. Jolley, “Neuromorphic Event-based Sensors for Machine Vision Applications,” presented at the Air and Space Power Centre Confrence, 2022. Accessed: Jun. 13, 2024. [Online]. Available: https://www.youtube.com/watch?v=9SbftS0V6Sk&ab_channel=AirandSpacePowerCentre

[47] J. P. Wachs, M. Kölsch, H. Stern, and Y. Edan, “Vision-based hand-gesture applications. Commun ACM,” vol. 54, pp. 60–71.

[48] M. Calum, “Signapse’s Sign Language Translation using Generative AI: Achieving Photo-Realism and Accuracy,” 2023.

[49] X. Zhao, A. M. Naguib, and S. Lee, “Kinect based calling gesture recognition for taking order service of elderly care robot,” presented at the The 23rd IEEE International Symposium on Robot and Human Interactive Communication, 2014, pp. 525–530. doi: https://doi.org/10.1109/ROMAN.2014.6926306.

[50] J. Falcón, “ADAM: Meet the multipurpose professional assistance robot for homes.” Accessed: Jan. 25, 2024. [Online]. Available: https://inspenet.com/en/noticias/meet-the-robot-multipurpose-home-robot/

[51] A. Gies, “Kinect Sports Review. Rare’s Kinect Debut redefines motion-controlled sports.,” IGN. [Online]. Available: https://www.ign.com/articles/2010/11/04/kinect-sports-review

[52] M. Montoya, J. F. Villada, J. E. Muñoz, and O. Henao, “Exploring Coordination Patterns in VR-Based Rehabilitation for Stroke Using the Kinect Sensor,” in *HCI in Games: Serious and Immersive Games*, 2021, pp. 372–387. doi: 10.1007/978-3-030-77414-1_27.

[53] “FALCON NEURO,” afresearchlab. [Online]. Available: https://afresearchlab.com/technology/falcon-neuro/

[54] Tecnologico de Monter, “Azure Kinect DK Specifications (Skeletal Tracking),” *IFE Experiential Classroom*, 2022. [Online]. Available: https://ifelldh.tec.mx/sites/g/files/vgjovo1101/files/Azure%20Kinect%20DK%20Specifications.pdf

[55] R. Tchantchane, H. Zhou, S. Zhang, and G. Alici, “A Review of Hand Gesture Recognition Systems Based on Noninvasive Wearable Sensors,” vol. 5, no. 10, 2023, doi: 10.1002/aisy.202300207.

[56] S. F. Khan, N. H. Mohd, D. M. Soomro, S. Bagchi, and M. D. Khan, “3D Hand Gestures Segmentation and Optimized Classification Using Deep Learning,” vol. 1, no. 1, p. 99, doi: 10.1109/ACCESS.2021.3114871.

[57] B. Gaudenz, “MediaPipe: Google’s Open Source Framework (2024 Guide).” Accessed: Mar. 24, 2024. [Online]. Available: https://viso.ai/computer-vision/mediapipe/

[58] L. Chandwani *et al.*, “Gesture based Sign Language Recognition system using Mediapipe.” 2023. doi: 10.21203/rs.3.rs-3106646/v1.

[59] D. Pogue, “Leap Motion Controller, Great Hardware in Search of Great Software,” 2013. Accessed: Mar. 19, 2024. [Online]. Available: https://www.nytimes.com/2013/07/25/technology/personaltech/no-keyboard-and-now-no-touch-screen-either.html

[60] G. Krastev and M. Andreeva, “A SOFTWARE TOOL FOR EXPERIMENTAL STUDY LEAP MOTION,” *IJCSIT*, vol. 7, no. 6, 2015, doi: 10.5121/ijcsit.2015.7612.

[61] “Intel RealSense Product Family D400 Series Datasheet,” *Intel Corporation*, p. 55, 2020.

[62] “RealSense Depth Camera D415 Technical Specifications,” Intel Corporation. Accessed: Mar. 21, 2024. [Online]. Available: https://store.intelrealsense.com/buy-intel-realsense-depth-camera-d415.html#tech-specs

[63] “RealSense Depth Camera D435 Technical Specifications,” Intel Corporation. Accessed: Mar. 21, 2024. [Online]. Available: https://store.intelrealsense.com/buy-intel-realsense-depth-camera-d415.html#tech-specs

[64] “RealSense Depth Camera D455 Technical Specifications,” Intel Corporation. Accessed: Mar. 21, 2024. [Online]. Available: https://store.intelrealsense.com/buy-intel-realsense-depth-camera-d455.html#tech-specs

[65] “Intel RealSense Product Family D400 Series Datasheet,” *Intel Corporation*, pp. 36, 37, 52, 2020.

[66] “Intel RealSense Product Family D400 Series Datasheet,” *Intel Corporation*, p. 36, 2020.

[67] D. Abramson, “Robotic Automation for Smart Agriculture and Renewable Energy,” 2021.

[68] M. I. Harris and A. S. Agoes, “Applying Hand Gesture Recognition for User Guide Application Using MediaPipe,” in *Advances in Engineering Research*, Atlantis Press International B.V, 2021. doi: 10.2991/aer.k.211106.017.

[69] G. Douglas W., “UGV History 101: A Brief History of Unmanned Ground Vehicle (UGV) Development Efforts,” NAVAL COMMAND CONTROL AND OCEAN SURVEILLANCE CENTER RDT AND E DIV SAN DIEGO CA, SAN DIEGO CA, ADA422845, 2004.