universität wien

MASTERARBEIT / MASTER'S THESIS

Titel der Masterarbeit / Title of the Master's Thesis

"A Comparative Analysis of Image Recognition Cloud Services"

verfasst von / submitted by

Ziad Al Sarrih

angestrebter akademischer Grad / in partial fulfilment of the requirements for the degree of

Master of Science (MSc)

Wien, 2024 / Vienna, 2024

Studienkennzahl lt. Studienblatt / degree programme code as it appears on the student record sheet:	UA 066 921
Studienrichtung lt. Studienblatt / degree programme as it appears on the student record sheet:	Masterstudium Informatik
Betreut von / Supervisor:	Univ. Prof. Dr. Wolfgang Klas

Abstract

The rapid advancements in computer vision technologies by industrial companies like Amazon, Google, IBM, and Microsoft have led to the development of robust image recognition APIs. These APIs have revolutionized the way programmers and businesses utilize computer vision, eliminating the need for building complex systems from scratch.

Although these APIs offer considerable advantages, they also present notable challenges in terms of performance, cost, limitations, and functionalities. It is essential to conduct an in-depth analysis and comparison of these APIs to assist developers and organizations in making well-informed choices. Selecting the most suitable API in accordance with specific requirements is a complex task. For instance, a minor error like misreading a car license plate number by an API can lead to serious repercussions, demonstrating the critical nature of accurate API performance in real-life scenarios.

The current APIs offer a wide range of services, including image classification, face recognition, image retrieval, optical character recognition, and handwriting interpretation. These APIs have democratized access to powerful Artificial Intelligence (AI) capabilities, enabling developers to focus on the pipeline aspects of their applications. In order to select the most suitable API that meets the specified requirements, the existing approach recommends reviewing the documentation of one or more APIs, or alternatively, conducting tests on them. However, these methods are time-consuming, resource-intensive, and may not be sufficiently reliable. Given the impracticality of thoroughly examining and testing all documentation of all the API providers, a more efficient and effective selection process is required.

This research aims to study and compare these computer vision APIs. It will assess their performance, cost, functionality, and limitations. Additionally, a prototype tool will be developed to assist in selecting the best API/s for specific circumstances, considering factors such as application type, budget constraints.

The goal of this study is to conduct an in-depth analysis and comparison of various computer vision APIs, evaluating aspects such as their performance, cost, functionality, and limitations. A decision support tool is developed to assist in select the most appropriate API for particular situations. This system will focus on four key areas: performance, costs, functionality, and limitations. The tool is designed to automatically gather and analyze API documentation from websites, utilizing a machine learning algorithm. Based on the user's specified criteria, such as cost or performance..., the tool is organizing and presenting the relative information for all chosen APIs in an ascending order based on an evaluation criteria.

In conclusion, the emergence of advanced computer vision APIs from industrial companies has significantly transformed the landscape of AI industry. These APIs, while offering immense benefits in terms of functionality and ease of use, also bring three challenges in

performance, cost and limitations. The necessity for a comprehensive comparison and analysis of these APIs is evident, as the correct choice is crucial to avoid potential risks and maximize efficiency. This research has aimed to address these needs by providing an in-depth evaluation of various computer vision APIs and developing a decision support system. This system, leveraging machine learning for analysis, simplifies the selection process by aligning API capabilities with user-specific requirements. Ultimately, this study not only contributes to a deeper understanding of the current API landscape but also offers decision support tools to assist developers and organizations in making informed decisions, thereby facilitating more effective and reliable application development in the field of computer vision.

Kurzfassung

Die schnelle Entwicklung in den Computer Vision Technologien von Industrieunternehmen wie Amazon, Google, IBM und Microsoft hat zur Entwicklung robuster Bilderkennungs-APIs geführt. Diese APIs haben die Art und Weise, wie Programmierer und Unternehmen Computer Vision nutzen, revolutioniert und die Notwendigkeit eliminiert, komplexe Systeme von Grund auf neu zu entwickeln.

Obwohl diese APIs erhebliche Vorteile bieten, präsentieren sie auch bemerkenswerte Herausforderungen in Bezug auf Leistung, Kosten, Einschränkungen und Funktionalitäten. Es ist wesentlich, eine tiefgehende Analyse und einen Vergleich dieser APIs durchzuführen, um Entwicklern und Organisationen zu helfen, gut informierte Entscheidungen zu treffen. Die Auswahl der am besten geeigneten API gemäß spezifischen Anforderungen ist eine komplexe Aufgabe. Beispielsweise kann ein kleiner Fehler, wie das Fehllesen einer Autokennzeichennummer durch eine API, ernsthafte Folgen haben und zeigt die kritische Natur der genauen API-Leistung in realen Szenarien.

Die aktuellen APIs bieten eine breite Palette von Dienstleistungen an, darunter Bildklassifizierung, Gesichtserkennung, Bildabruf, optische Zeichenerkennung und Handschrifterkennung. Diese APIs haben den Zugang zu leistungsstarken Künstlicher-Intelligenz-(KI)-Fähigkeiten demokratisiert, sodass Entwickler sich auf die Pipeline-Aspekte ihrer Anwendungen konzentrieren können. Um die am besten geeignete API auszuwählen, die die festgelegten Anforderungen erfüllt, empfiehlt der bestehende Ansatz, die Dokumentation einer oder mehrerer APIs zu überprüfen oder Tests an ihnen durchzuführen. Diese Methoden sind jedoch zeitaufwändig, ressourcenintensiv und möglicherweise nicht ausreichend zuverlässig. Angesichts der Unpraktikabilität, alle Dokumentationen aller API-Anbieter gründlich zu untersuchen und zu testen, ist ein effizienterer und effektiverer Auswahlprozess erforderlich.

Diese Forschung zielt darauf ab, diese Computer Vision APIs zu studieren und zu vergleichen. Sie wird ihre Leistung, Kosten, Funktionalität und Einschränkungen bewerten. Zusätzlich wird ein Prototyp-Tool entwickelt, um bei der Auswahl der besten API/s für spezifische Umstände zu helfen, unter Berücksichtigung von Faktoren wie Anwendungstyp und Budgetbeschränkungen.

Das Ziel dieser Studie ist es, eine tiefgehende Analyse und einen Vergleich verschiedener Computer Vision APIs durchzuführen, Aspekte wie ihre Leistung, Kosten, Funktionalität und Einschränkungen zu bewerten. Ein Entscheidungshilfesystem wird entwickelt, um bei der Auswahl der geeignetsten API für bestimmte Situationen zu helfen. Dieses System konzentriert sich auf vier Schlüsselbereiche: Leistung, Kosten, Funktionalität und Einschränkungen. Das Tool ist so konzipiert, dass es automatisch API-Dokumentationen von Websites sammelt und analysiert, unter Verwendung eines maschinellen Lernalgorithmus. Basierend auf den vom Benutzer angegebenen Kriterien, wie Kosten oder Leistung..., or-

ganisiert und präsentiert das Tool die entsprechenden Informationen für alle ausgewählten APIs in aufsteigender Reihenfolge basierend auf einem Bewertungskriterium.

Zusammenfassend hat das Aufkommen fortschrittlicher Computer Vision APIs von Industrieunternehmen die Landschaft der KI-Branche erheblich verändert. Diese APIs bieten zwar immense Vorteile in Bezug auf Funktionalität und Benutzerfreundlichkeit, bringen aber auch drei Herausforderungen in Bezug auf Leistung, Kosten und Einschränkungen mit sich. Die Notwendigkeit eines umfassenden Vergleichs und einer Analyse dieser APIs ist offensichtlich, da die richtige Wahl entscheidend ist, um potenzielle Risiken zu vermeiden und die Effizienz zu maximieren. Diese Forschung hat zum Ziel, diese Bedürfnisse zu adressieren, indem sie eine eingehende Bewertung

verschiedener Computer Vision APIs und die Entwicklung eines Entscheidungshilfesystems. Dieses System, das maschinelles Lernen für die Analyse nutzt, vereinfacht den Auswahlprozess, indem es die Fähigkeiten der API mit benutzerspezifischen Anforderungen abgleicht. Letztendlich trägt diese Studie nicht nur zu einem tieferen Verständnis der aktuellen API-Landschaft bei, sondern bietet auch Entscheidungshilfetools, um Entwicklern und Organisationen bei der fundierten Entscheidungsfindung zu helfen, wodurch effektivere und zuverlässigere Anwendungsentwicklungen im Bereich der Computer Vision erleichtert werden.

Contents

List of Tables

List of Figures

1. Introduction

1.1. Introduction

Over the last several years, machine learning, and more specifically Deep Learning, have emerged as revolutionary technologies at the forefront of many people's minds [3]. This captivating development is remarkable when viewed through a historical lens, considering that research on artificial intelligence (AI) began in the mid-20th century, promising much but initially delivering little by the end of the century [34, 5]. It's not that the study was uninteresting or lacked potential; it simply didn't lead to the groundbreaking advances in areas like voice and sight recognition as initially envisioned.

Several factors hindered progress in early AI research, with one of the most significant being the complexity of both visual and auditory identification. These tasks, which seem intuitive to humans, proved exceedingly challenging to replicate in machines. Decades of research in traditional AI approaches, such as rule-based systems and expert systems, failed to deliver on the promises of creating intelligent machines.

However, around 2010, the implementation of neural networks provided the long-awaited breakthrough [34]. Neural networks, a concept inspired by the functioning of the human brain, had been around for decades but were often considered a failed attempt at solving AI challenges. Surprisingly, they re-emerged as the key to unlocking the potential of AI and machine learning, even when many in the academic world had shifted their attention away from them.

Since 2010, there has been a remarkable surge in research and development efforts focused on deep learning, resulting in a steady stream of articles in the technical press highlighting its wide-ranging successes across various domains. Major technology companies such as Amazon, Apple, Facebook, and Google have recognized the transformative potential of deep learning and devoted massive resources to advance this technology [5]. Consequently, modern technology has seen extensive integration of deep learning algorithms into various applications, enabling significant advancements in areas like computer vision, natural language processing, and speech recognition.

Smartphones, which have become ubiquitous in our lives, now incorporate deep learning models that can recognize individual faces and accurately interpret spoken instructions. The integration of such capabilities into our handheld devices showcases the real-world impact of deep learning on our daily lives. Moreover, the future of autonomous vehicles, once a distant dream, is now on the cusp of reality, largely due to advancements in deep learning algorithms [58].

The practical applications of deep learning extend far beyond consumer electronics. Industries such as healthcare, finance, transportation, and manufacturing are leveraging

this technology to enhance efficiency, accuracy, and decision-making processes [54]. For instance, in healthcare, deep learning is revolutionizing medical imaging, enabling early detection of diseases and improving patient outcomes. Despite these significant advancements, challenges accompany the widespread adoption of deep learning. Issues related to data privacy, bias in AI systems, and the potential impact on the job market are among the critical areas that demand attention [58].

AI's progress, from its early beginnings in the twentieth century to the revolutionary benefits of deep learning in recent years, demonstrates the power of perseverance and ingenuity. With the resuscitation of neural networks and subsequent developments in deep learning, a new era of technological growth has begun, significantly changing numerous sectors and the way people engage with technology. To ensure that AI benefits society as a whole while limiting its drawbacks, researchers and developers must address these concerns responsibly as they explore the potential of deep learning. Without a doubt, deep learning has a bright future and the potential to open up even more astounding possibilities in the coming years.

Image recognition, also known as picture analysis in computer science, employs a number of algorithms and machine learning concepts to recognize the environments, people, and activities depicted in photographs [26]. Deep learning has refined this technique dramatically in recent years, allowing computers to detect visual and even aural signals in images. One common use of image recognition is through image recognition APIs (Application Programming Interfaces). These APIs make it straightforward to conduct picture recognition tasks without getting mired down in the complexities of constructing models from scratch by allowing developers to include pre-trained models and algorithms into their apps.

For example, the Microsoft Azure Face API can derive vital facts about a person's age, sex, and emotional state by evaluating a photo of their face [26]. This API delivers precise insights on several facial traits by utilizing cutting-edge machine learning models and face recognition techniques. Another example is the IBM Watson Visual Recognition API, which excels in identifying the categories of items in a photograph. Watson Visual Recognition detects furniture, appliances, and other objects in images using machine learning algorithms, allowing users to identify and grasp the content of their visual data.

As image recognition APIs have grown in popularity, it has become critical to provide a framework for comparing them to identify important differences and best practices. A framework like this would let programmers and businesses choose the optimal API for their specific use cases. The study's main purpose is to evaluate several aspects of image recognition APIs, such as performance, cost, functionality, and limitations [57].

Developers must also consider functionality because it may significantly reduce the time and effort required to use the API in their projects. Users may make decisions based on their budget and demands by reviewing price models to determine the cost implications of using various image recognition APIs [2]. Furthermore, the study would explore the APIs' capabilities in recognizing the visual cues, if applicable. With the advancements in deep learning, some APIs might excel in understanding only the visual content of an image but also audio elements if applicable.

Image recognition APIs have revolutionized the way we interact with images, enabling applications to decipher complex visual information effortlessly [3]. By presenting a comprehensive framework for comparing image recognition APIs, this study aims to assist developers and businesses in selecting the API that aligns best with their specific needs. As the field of image recognition continues to evolve, such evaluations will prove instrumental in advancing the state-of-the-art and driving further innovation in this exciting domain.

The development of deep learning has revolutionized the field of image recognition, propelling it forward alongside powerful artificial intelligence. Thanks to these advancements, image categorization has seen significant progress, and there is potential for real-time object detection systems, including face recognition, to surpass human performance in both accuracy and speed [9]. As we rely on image recognition APIs, finding a balance between speed and resource conservation becomes crucial in addressing the challenges of computer vision in an affordable and efficient manner.

The key to achieving this balance lies in the optimal combination of deep learning algorithms with hardware and software designed specifically for their utilization. When it comes to computer vision tasks, having access to the latest and most powerful algorithms can yield substantial financial benefits [57]. These advanced algorithms are not only more productive compared to their predecessors but also leverage technology that is significantly more cost-effective, offering superior performance on the same hardware infrastructure.

The evaluation of existing image recognition APIs is paramount in determining which one is best suited for a particular task. These APIs serve a multitude of purposes across various industries, including retail, safety, and social media. By incorporating these APIs into their applications, developers can effortlessly and rapidly integrate powerful image recognition features without the need to build and train their own models from scratch [13]. However, the plethora of picture recognition APIs available in the market presents developers with an array of choices, each offering its own set of advantages and drawbacks concerning performance, cost, functionality, limitations, and more.

In the retail industry, image recognition APIs have proved invaluable in enabling businesses to enhance customer experiences. For instance, by integrating image recognition capabilities into their mobile applications, retailers can offer features such as product recognition, allowing customers to snap a photo of an item and instantly find it within their inventory. Additionally, image recognition assists in providing personalized product recommendations based on the user's preferences and past purchase history. In the realm of safety and security, image recognition APIs are deployed to monitor and analyze surveillance footage in real-time. These APIs can identify and alert security personnel to potential threats, suspicious activities, or unauthorized access, thus bolstering the effectiveness of surveillance systems and reducing response times in critical situations.

Moreover, social media platforms heavily rely on image recognition APIs to analyze and categorize the vast amounts of visual content uploaded by users every day. This technology enables automatic tagging of images, making it easier for users to search for specific content and facilitating targeted advertising based on the visual elements present in users' posts.

However, selecting the most suitable image recognition API for a specific application requires careful consideration. Factors like the complexity of the recognition task, the desired level of accuracy, real-time processing requirements, and budget constraints must all be taken into account during the evaluation process.

Deep learning's rapid advancement has brought image recognition to new heights, enabling real-time object detection and potentially surpassing human performance in certain tasks. The availability of image recognition APIs has accelerated the integration of this technology across multiple sectors by allowing developers to construct apps with powerful picture identification capabilities without the need for time-consuming model training. As the need for image recognition grows, a rigorous evaluation of current APIs is critical in choosing the optimal solution for specific activities and use cases. By exploiting the capabilities of these APIs, businesses may open new doors for expansion, efficiency, and enhanced user experiences.

1.2. Background

Because of the significant progress made by technical behemoths such as Amazon, Google, IBM, and Microsoft, a wide range of computer vision APIs with identical characteristics are now readily available. These APIs allow programmers and businesses to leverage computer vision without having to build complex systems from the ground up [17]. Text reading, handwriting analysis, picture similarity detection, face and celebrity identification, and photo tagging are just a few of the features they provide. The convenience of these APIs lies in their user-friendly interfaces, allowing developers to quickly integrate computer vision capabilities into their applications without extensive coding requirements.

One remarkable aspect of these tech giants' efforts is their continuous commitment to enhancing computer vision technologies [19]. They gathered and meticulously categorized massive datasets with millions of dollars invested, exceeding the legendary ImageNet collection's accuracy. These labeled datasets serve as important training material for machine learning models, improving the accuracy and reliability of the APIs' outputs.

The fact that these APIs are cloud-based adds to their appeal. Because of its ease of use and rapid deployment, developers may get started on their computer vision projects immediately, reducing the time it takes for new applications to reach the market [18]. The APIs contain a wide variety of features and a large number of tags and annotations to support various image recognition tasks. Another significant advantage of cloud-based computer vision APIs is their low cost. Instead of hiring a costly data science team, these APIs allow firms to leverage cutting-edge computer vision technologies at a low cost. This democratizes access to cutting-edge AI capabilities, making them available to a broader range of organizations and individuals.

With these APIs, programmers can focus on the unique aspects of their apps while leaving the difficult computer vision algorithms and model training to the pros. This simplicity and ease of use serve to explain why computer vision technologies are employed in so many different areas of the business. The ongoing advancements made by these digital giants are hastening the development of the computer vision environment [20].

API performance and accuracy have been improved, enabling more sophisticated real-time applications. Now, the APIs are able to grasp complex scenarios, identify minute details, and even interpret emotions from facial expressions.

The computer vision APIs from Amazon, Google, IBM, and Microsoft have revolutionized image analysis and recognition. Their ease of use, diverse capabilities, rich dataset labels, and reasonable pricing make them indispensable tools for developers and businesses alike. As these tech giants continue to invest in the advancement of computer vision technologies, we can expect even more remarkable applications and innovations in the near future [26]. The democratization of AI through these APIs enables organizations of all sizes to harness the power of computer vision and drive progress across various domains.

In recent years, there has been a lot of emphasis placed on evaluating and contrasting various image recognition APIs. Researchers and developers are most interested in finding the most effective and efficient APIs for their particular applications [57, 21]. Studies frequently compare the performance of various APIs using well-known datasets like ImageNet and COCO to determine their accuracy and capability.

While accuracy is a crucial consideration, it is essential to recognize that it is just one factor among many that influence the choice of an API. Other important factors include pricing, available features, and any applicable constraints specific to the project. The cost of image recognition APIs can vary widely depending on the supplier and the application's demands. Some service providers charge their clients per service request, while others offer different pricing tiers based on the volume of requests made monthly [22]. Moreover, certain providers may extend discounted pricing or special deals for public benefit organizations like schools and hospitals. For businesses, it becomes imperative to formulate a strategy that incorporates image recognition APIs into their products or services at the most affordable cost.

Furthermore, the development of image recognition technology relies on cooperation between the AI research community, decision-makers, and industrial stakeholders. Open discussions and group efforts are essential to address the difficulties and possible threats connected with AI applications, including image recognition APIs [23].

When selecting the best tool for a given application, evaluating and comparing image recognition APIs is essential. Performance, cost, functionality, and limitations are all significant factors that need to be carefully weighed during the selection process. Even though these APIs enable previously unheard-of capabilities and convenience, it is critical to be conscious of the privacy risks that come with their widespread use [27]. By adopting responsible practices and fostering open dialogues, we can harness the power of image recognition technology and ensure fair and inclusive AI methods.

The growing adoption of artificial intelligence in decision-making processes raises concerns about performance, cost, functionality, and limitations [57]. In some cases, large organizations and AI service providers may rush the adoption of face analysis technology without conducting thorough algorithm testing to save time and resources [26]. An example of this is the Oregon County Sheriff's Office in the US, which has implemented Amazon's Rekognition software. However, research has revealed alarming inaccuracies,

with 28 US senators and representatives being wrongly associated with arrestees' mugshots by the software [2]. Even more concerning is the fact that people of color, who make up only 20% of Congress, accounted for nearly 40% of the Rekognition's wrongful matches [2]. Such inaccuracies can lead to wrongful arrests and police harassment, making it imperative to carefully assess the performance and accuracy of available image recognition APIs before making a choice [58].

When considering image recognition APIs, it is crucial to evaluate their features and constraints. Some APIs may offer a wide range of capabilities, while others may be more specialized in tasks like facial or object recognition. Additionally, the image quality and range of emotions that an API can process might also be limiting factors [5]. Developers and organizations need to align their specific requirements with the capabilities offered by different APIs to ensure a suitable match.

However, choosing the right image recognition API is not always straightforward, as reliable and up-to-date information about various APIs may be scarce. This lack of information can make it challenging for software engineers and company executives to make informed decisions. To address this issue, extensive research that evaluates and compares key image recognition APIs based on performance, cost, functionality, and limitations would be highly beneficial for developers and consumers alike [31]. Transparency and accountability are essential when integrating AI technologies like image recognition into real-world applications. Service providers and developers must prioritize testing and validation processes to ensure performance and accuracy. Periodic audits and reviews help assure the reliability of AI systems.

The expanding use of image recognition APIs and AI decision-making has raised concerns about four categories: performance, cost, functionality, and limitations. Incorrect matches and possible injustices serve as a harsh reminder of the significance of properly assessing image recognition APIs before usage [50]. Organizations must carefully analyze the features and limitations of each API to ensure that they satisfy their specific requirements. Developers and organizations would benefit greatly from research that provides comprehensive insights into the performance, cost, functionality, and limitations of various APIs. By fostering openness, accountability, and cooperation, we can build a responsible and reliable AI ecosystem.

1.3. Research Aim

The goal of this master's thesis is to meet the need for a comprehensive examination and comparison of well-known image recognition APIs. The research will focus on examining four categories: performance, cost, functionality, and limitations of various APIs in order to assist developers and companies in making informed recommendations for each API. The thesis aims to shed light on the benefits and drawbacks of APIs to aid users in deciding which API best meets their specific needs.

The thesis also aims to prototypically implement a comparison support tool that helps compare the four categories between APIs. This tool will be valuable in providing users with informative information about which API would be most suited for their project

according to selected categories. The goal is to provide developers and organizations with a simple and effective way to align the best image recognition API, taking into consideration the four categories: performance, cost, functionality, and limitations.

Through this research and the development of the prototype tool, the master's thesis aims to advance image recognition technology in a range of areas by providing users with the knowledge and tools they need.

1.4. Research Questions

1. What are the accuracy issues with respect to the use of these APIs? To answer this, we will look at possible issues of mistakes, including cases where images were not properly recognized (see Section 4.2.1).

2. What is the performance of major image recognition APIs? Detailed information on the APIs and the number of images they can accurately process is important to answer this question (see Section 4.2.1).

   Performance Metric: The performance metric focuses on evaluating the accuracy and efficiency of major image recognition APIs. To measure performance, consider two important aspects:

   - Accuracy: Ability to correctly recognize and classify images. API's precision, recall, and overall classification accuracy. Information about the APIs' accuracy rates and their performance on specific image datasets is crucial for this analysis.
   - Efficiency: Speed and scalability of the APIs. The time taken by the APIs to process a given number of images or requests, as well as their ability to handle large-scale image processing tasks effectively.

3. What are the cost implications? Major APIs are subscription-based and, hence, subscription costs may increase with the number of images processed. It is important to show how the cost could impact the usage of APIs (see Section 4.2.2).

   Cost Metric: The cost metric focuses on the financial implications associated with the use of image recognition APIs. To evaluate the cost implications, consider the following factors (Table 4.1):

   - Subscription-based Pricing: Major APIs typically employ a subscription-based pricing model. We aim at assessing the cost structure and subscription plans offered by each API provider, including any tiered pricing based on the number of images processed or additional features.
   - Scalability Costs: Costs increase as the number of images processed or API usage scales up. Identify any potential cost limitations that could impact the affordability or feasibility of using these APIs for large-scale image recognition tasks.

4. What is the level of functionality of APIs? Range of capabilities and features (see Section 4.2.3).

Functionality Metric: The functionality metric assesses the range of capabilities and features offered by image recognition APIs. Consider the following aspects:

- Recognition Capability: Functionalities provided by the APIs and determine the breadth and depth of the recognition capabilities provided by each API, such as facial recognition, scenes, object detection, image tagging, or sentiment analysis.

5. What are the limitations and constraints? Technical and usage restrictions (see Section 4.2.4).

   Limitation Metric: This metric focuses on identifying any limitations or constraints associated with the use of image recognition APIs. Consider the following factors:

   - Technical Limitations: Maximum file size, resolution, or format restrictions imposed by the APIs for image input. Constraints on the number of concurrent requests, rate limits, or throughput limitations that may affect the performance or scalability of the APIs.
   - Usage Restrictions: Terms of service, usage policies, or restrictions imposed by the API providers, such as limitations on sensitive image data, prohibited use cases, or restrictions on deploying the API in certain regions or industries.

1.5. Research Motivation

The rapid progress in artificial intelligence (AI) and machine learning technologies has led to significant transformations across various industries and daily activities. One of the most impactful applications of AI is image recognition, which enables machines to analyze and understand visual information much like the human visual system [3, 54]. This advancement has revolutionized the way we interact with technology and perceive the world around us.

A key driving force behind the widespread adoption of image recognition technology is the availability of image recognition APIs (Application Programming Interfaces). These APIs allow developers and businesses to seamlessly integrate powerful image recognition capabilities into their products and services without the need to develop complex models from scratch. The APIs offered by major tech companies such as Google, Microsoft, Amazon, and IBM provide access to state-of-the-art AI algorithms and resources, democratizing the use of image recognition systems and making them more accessible to a wider range of developers.

Image recognition APIs' versatility has broadened the industries in which they are employed. Face recognition APIs can accurately identify and validate persons, improving security and speeding up user authentication operations. Image recognition is significantly used in medical imaging to aid in the early detection and diagnosis of a wide range of medical conditions. The automobile industry uses image recognition technology to help autonomous vehicles comprehend their surroundings and make sound decisions

[21]. Furthermore, image recognition APIs have empowered merchants and e-commerce enterprises. By identifying products and objects, these APIs enable augmented reality experiences, allowing customers to virtually try on items before making a purchase. This enhances the whole shopping experience and helps businesses increase revenue.

The availability of image recognition APIs has also fueled innovation in sectors such as sentiment analysis on social networking platforms, visual search, and content moderation. These APIs enable automatic photo tagging and classification, improving user experience and making it easier for users to obtain relevant content. Photo recognition APIs are evolving and improving over time [31]. Tech companies invest significantly in improving and expanding the capabilities of their APIs, resulting in greater accuracy, better performance, and improved compatibility for a wide range of image types and circumstances.

However, despite the numerous benefits, the use of image recognition APIs also raises some concerns related to privacy and reliability. As these APIs become more integrated into our daily lives, it becomes imperative for developers and organizations to approach their implementation responsibly and address potential issues related to fairness and user privacy [23]. Image recognition APIs have become pivotal in the advancement of AI and machine learning technologies. They offer developers and businesses access to powerful image recognition capabilities without the need for extensive expertise in AI modeling. The versatility of these APIs has enabled applications across a wide range of industries, from security and healthcare to e-commerce and social media. As these APIs continue to evolve, it is essential to consider the performance and accuracy of their use and work towards building fair, inclusive, and reliable AI systems that benefit society as a whole.

The rapid growth of the image recognition API market has brought forth challenges for developers and decision-makers. Selecting the most suitable API for specific use cases is vital, and developers must consider a myriad of factors, including performance, cost, functionality, limitations, and more [9]. Understanding the trade-offs between different APIs is essential to improve application performance, manage costs efficiently, and utilize AI technology responsibly.

The purpose of this research is to provide developers, corporations, and researchers with a comprehensive and comparative analysis of image recognition APIs. This master's thesis aims to address this gap by evaluating and comparing the accuracy, costs, functionality, and limitations of major image recognition APIs. Additionally, it aims to prototypically implement a comparison support tool that compares APIs based on the four categories.

The assessment and comparison of image recognition APIs in the study are helpful tools for the IT community, guiding programmers through the complex API environment. By studying the benefits and drawbacks of each API, developers may make decisions that are more in line with the demands and aims of their projects [18]. Furthermore, this study lends credence to the case for AI adoption. By taking proactive steps to address the accuracy implications of image recognition APIs, developers and decision-makers may avoid bad results and protect user privacy. To employ AI technology for the betterment of society while avoiding potential hazards and bad consequences. The outcomes of the study might also assist corporations and organizations in improving their AI endeavors.

By selecting the best image recognition API, businesses may improve the efficiency and efficacy of their AI-driven apps, resulting in improved user experiences and commercial outcomes.

As image recognition APIs expand, developers and decision-makers encounter both opportunities and challenges. Making informed judgments that are in accordance with project objectives necessitates a thorough examination of four categories: performance, cost, functionality, and limitations. This study aims to provide a comprehensive and comparative examination of image recognition APIs, giving useful insights for programmers, companies, and researchers [27]. By understanding the trade-offs of various APIs, developers may employ AI technology effectively, culminating in the production of innovative AI apps that benefit society.

Because AI technology is always growing, it is critical to stay up to date with the most recent breakthroughs in image recognition APIs. Developers will be able to stay up with the quickly changing business environment and fully harness the potential of AI-powered image analysis with the help of this research [57]. The ultimate purpose of this study is to provide companies and programmers with the tools they need to use image recognition APIs in a useful manner. This research encourages the appropriate use of AI technology by enabling informed decision-making through a thorough assessment and comparison of various APIs. As a result, it is envisaged that AI-powered image identification would become more extensively utilized, driving the creation of new applications and advancements in a variety of fields.

This study ensures that developers are ready to leverage the capabilities of AI-driven image analysis in their projects by keeping them up to speed on the most recent advances and trends in image recognition APIs. By addressing the four categories' challenges, the research hopes to build a culture of responsible AI adoption, in which technology is used for societal benefit while avoiding potential hazards and bad consequences. This study aims to advance and promote the usage of AI-based image recognition. By providing relevant information and insights, it helps organizations and developers to employ AI technology in a helpful, and forward-thinking manner. AI-powered picture recognition has the potential to transform numerous industries and result in revolutionary advances in technology and beyond via suitable application and continued innovation.

1.6. Summary

Chapter one highlights the significance of machine learning and deep learning in modern culture, emphasizing their widespread use in various fields like online platforms and autonomous vehicles. It provides historical insights into the evolution of AI and deep learning, highlighting significant milestones and breakthroughs in recent times.

The introduction also explores the accessibility of computer vision application programming interfaces (APIs) provided by well-known technological companies. The use of these APIs has helped democratize access to advanced image recognition capabilities, making them more accessible to developers and businesses. Nonetheless, the discussion goes beyond accessibility to address critical issues such as the reliance on large labeled

datasets, the advantages of cloud-based APIs, and financial implications. The study intends to thoroughly investigate and compare image recognition APIs, focusing on their performance, cost-effectiveness, usefulness, and limitations. Furthermore, the goal of this project is to develop a prototype tool that compares APIs based on four selected categories.

Within its scope, the paper specifies four categories: performance, cost, functionality, and limitations. The aforementioned questions serve as core concepts that guide the thesis's further research and analysis. The chapter's final part emphasizes the significant and far-reaching implications of artificial intelligence (AI) and image recognition in various industries. The first chapter lays the groundwork for the subsequent sections, which will delve further into the numerous components of this research, providing important views on the evolution of machine learning and deep learning algorithms within the field of image recognition.

1.7. Outline of the Master Thesis

This section provides a short overview of each chapter, summarizing what can be found in each one.

Chapter 1 provides an overview of the research topic, including its introduction, background, research aim, research questions, and motivation.

Chapter 2 discusses the use of Google Cloud Vision API, Amazon Rekognition, Microsoft Azure Computer Vision API, Clarifai, and Imagga, their research significance, recent developments, and the workings of CNNs. It also covers the connectivity of these networks, their input and output relationships, and the biases in datasets. The review also discusses the deep learning advances and challenges of these APIs and related work.

Chapter 3 discusses the methodology, which involves data collection, processing, vectorization using the fastText/facebook model, tokenization, clustering, and implementation of a comparison support tool, along with the data collection mechanism.

Chapter 4 explores image recognition API performance, accuracy, precision, recall levels, and considerations for choosing an API. It also covers cost implications, pricing models, subscription plans, language, platforms, image resolution, user interface, and additional features.

Chapter 5 provides a brief summary of each section, offers insights into the results, and concludes with key findings of the thesis.

Chapter 6 outlines potential future work, proposing enhancements to the current comparison support tool through the creation of advanced technologies such as a well-trained Chatbot Model.

2. Literature Review

2.1. Overview of Selected APIs

This section presents a comprehensive overview of selected image recognition APIs: Google Cloud Vision API, Amazon Rekognition, Microsoft Azure Computer Vision API, IBM Watson Visual Recognition, Clarifai, and Imagga. Each of these APIs represents a unique confluence of machine learning algorithms and practical functionality, offering a spectrum of services ranging from object detection and facial recognition to scene interpretation and text extraction.

2.1.1. Google Cloud Vision

Google achieved a huge advance in image recognition in 2014 when it won the ImageNet Large Scale Visual Recognition Competition (ILSVRC) competition using the 22-layer GoogLeNet model. This triumph paved the path for the now-common Inception architectures, which have become a cornerstone of current deep learning [18]. On the heels of this achievement, Google launched a set of Vision APIs in December 2015, giving developers significant tools for incorporating picture recognition capabilities into their apps. One of the key strengths of Google's Vision APIs is the wealth of user data it leverages for improving its classifiers. The large volume of data available to Google's deep learning models allows them to continuously improve their performance and accuracy. For instance, the knowledge gained from analyzing Google Street View images can be applied to real-world applications, such as text recognition on billboards, leading to robust and reliable outcomes.

Google's Vision APIs' facial recognition skills are particularly impressive. They offer the most comprehensive collection of face key points, including roll, tilt, and pan, which help in identifying numerous facial traits (Figure 2.1). Developers may use this degree of precision to create complex apps that need accurate facial analysis and monitoring.

Furthermore, Google's Vision APIs allow you to access online pictures that are visually comparable to the supplied image. Developers may acquire useful insights into how Google's image recognition engine analyzes and categorizes photographs by submitting photos to Google Photos and analyzing the accompanying tags. This feature allows developers to easily grasp the inner workings of Google's system without the need for substantial code.

By making these powerful Vision APIs accessible to developers, Google has democratized access to advanced image recognition technologies. Developers across various industries can now leverage the capabilities of these APIs to create innovative and intelligent applications. Whether it's in the fields of e-commerce, healthcare, or entertainment, Google's Vision

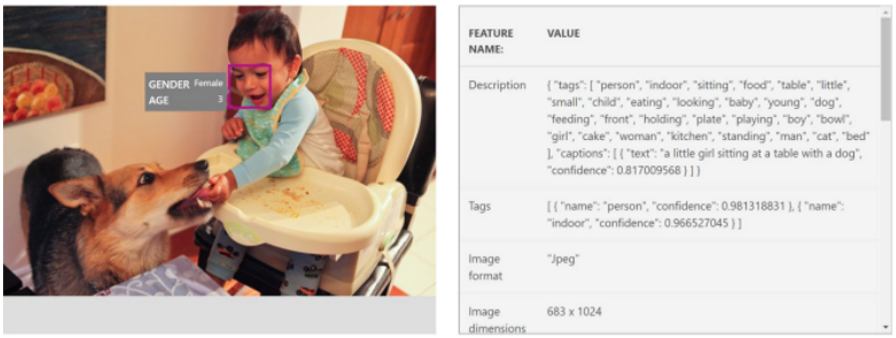

Figure 2.1.: ImageNet Recognition Competition shows the aid of the 22-layer GoogLeNet identified a young boy child interacting with a dog, tagged with concepts like "indoor," "sitting," and "food,"[55]

APIs offer a versatile and reliable solution for incorporating image recognition into a wide range of use cases.

Google's commitment to improving its image recognition technologies through the use of vast user data and continuous research ensures that its Vision APIs remain at the forefront of the industry. By providing developers with easy-to-use tools and powerful features, Google empowers them to unlock the full potential of AI-driven image analysis in their applications.

Google's success in image recognition with the GoogLeNet model and the subsequent release of the Vision APIs have revolutionized the field of image recognition. These APIs offer developers access to cutting-edge deep learning capabilities and a wealth of user data for improving classifiers. The comprehensive facial recognition features and the ability to retrieve visually similar images make Google's Vision APIs a valuable resource for developers seeking to integrate image recognition into their applications. As Google continues to invest in research and development, its Vision APIs are expected to drive further advancements and innovations in the field of image recognition.

2.1.2. Amazon Rekognition

Orbeus, a Sunnyvale, California-based firm, was purchased by Amazon in late 2015 and serves as the foundation for most of the Amazon Rekognition API (Figure 2.2). Orbeus, which was founded in 2012, rose to prominence after its principal scientist submitted winning bids to the 2014 ILSVRC detection competition. PhotoTime, a popular photo management tool, also used the company's APIs. These APIs are provided by Amazon Web Services (AWS) as part of its comprehensive portfolio [54]. Given the increasing number of companies providing image analysis APIs, Amazon has shifted its focus towards developing video recognition capabilities to differentiate itself in the market. As a result, Amazon Rekognition now offers notable enhancements, such as improved end-to-end integration with various AWS products, including Kinesis Video Streams, Lambda, and others. This integration allows developers to seamlessly incorporate Rekognition APIs

into their video streaming and processing workflows, providing a comprehensive and streamlined solution for video analysis [26].

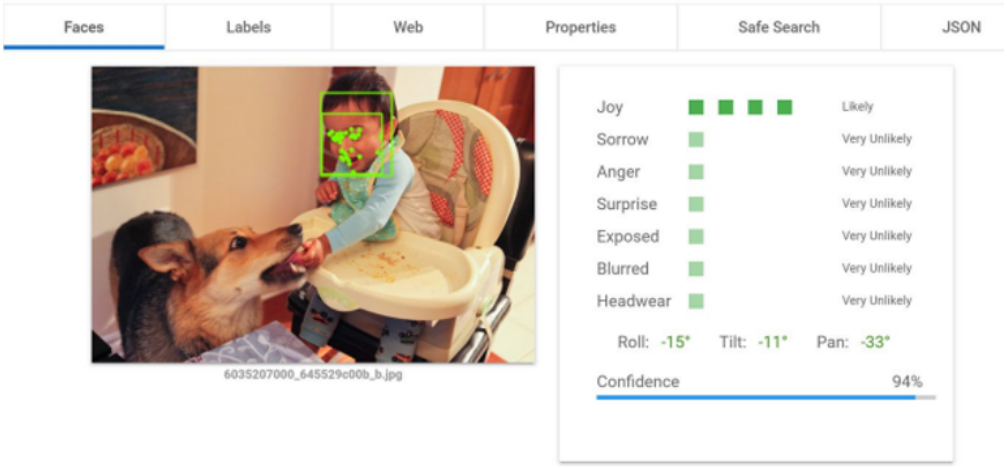

Figure 2.2.: Amazon Rekognition accurately identifies the joy on a child's face as they interact with their dog, with a high confidence level of 94%. The analysis picks up subtle cues for a joyful expression, while other emotions such as sorrow, anger, and surprise are very unlikely, according to the AI's algorithms.[55]

Amazon's commitment to innovation is evident through its introduction of additional APIs for specialized tasks such as license plate identification and video recognition. These APIs expand the scope of applications and use cases that developers can address using Amazon's AI capabilities.

A unique capability of the Amazon Rekognition API is its ability to determine whether the subject's eyes are open or closed. This feature can be particularly valuable in various applications, such as driver monitoring systems or facial expression analysis, where identifying the state of the subject's eyes can provide critical insights.

By leveraging the expertise and technology from Orbeus, Amazon has established itself as a major player in the image recognition and video analysis space. The integration of these capabilities into AWS offerings further solidifies Amazon's position as a comprehensive provider of AI-driven services for developers and businesses.

Amazon's acquisition of Orbeus has proven instrumental in the development of the Amazon Rekognition API, which offers powerful image recognition and video analysis capabilities. As part of AWS, these APIs are now accessible to a wide range of developers, enabling them to create innovative and sophisticated applications that leverage AI technology. With a focus on video recognition and continuous innovation, Amazon remains at the forefront of the image and video analysis industry, providing unique and valuable features that set its API apart from the competition.

2.1.3. Microsoft Azure Computer Vision

During the ILSVRC, the COCO picture captioning competition, and the Emotion Recognition in the Wild competition in 2015, Microsoft displayed its skill in a variety of tasks [13]. The exceptional performance across tasks, which included image classification, object detection, and generating visual descriptions, was attributed to the development of ResNet-152, a powerful deep learning model. Since then, research focus has shifted towards cloud-based APIs, and Microsoft took a significant step by introducing Project Oxford in 2015, which was later renamed Cognitive Services in 2016 [5]. This platform now offers a comprehensive suite of over 50 APIs covering diverse topics such as voice recognition, search functionalities, knowledge graph linking, vision capabilities, and natural language processing.

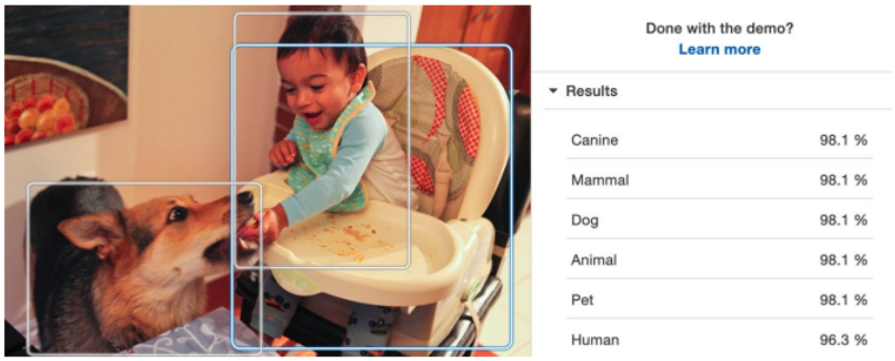

Figure 2.3.: Microsoft Azure Computer Vision's object detection confidently identifies the subjects in the image, distinguishing between human and animal with high precision. The image shows a child, recognized as 'Human' with 96.3% confidence, and a dog, categorized under 'Canine', 'Mammal', 'Dog', 'Animal', and 'Pet', each with a 98.1% confidence score, demonstrating the algorithm's robust capability in understanding and classifying different entities within a single scene.[55]

Initially, Microsoft's APIs were confined to use within the Xbox and Bing divisions, but they have since been democratized, extending accessibility to developers beyond the Microsoft ecosystem [17]. This democratization has unlocked novel opportunities for developers to integrate Microsoft's cutting-edge technologies into their applications, catalyzing creativity and innovation across diverse domains.

Several standout applications exemplify the inventive utilization of these APIs by developers. "How Old Do I Look?" (How-old.net) garnered attention, demonstrating Microsoft's face recognition capabilities as users uploaded photos to receive estimated age predictions. Another noteworthy application, the "Mimicker Alarm," showcases API versatility by requiring users to execute precise facial expressions to silence the morning alarm, underscoring the interactive and engaging potential of the APIs [2]. Furthermore, "CaptionBot.ai" gained popularity for generating descriptive captions for uploaded images, highlighting the potential of Cognitive Services' image captioning feature.

The Cognitive Services API encompasses a wide range of features, including image captioning, handwriting recognition, and headgear identification. It's crucial to emphasize Microsoft's commitment to user privacy and data protection [75]. Cognitive Services does not leverage consumer photo data to enhance its services, given its extensive clientele, including commercial clients relying on the platform for critical applications.

By affording developers access to potent AI capabilities through Cognitive Services, Microsoft has played a pivotal role in propelling AI adoption and nurturing a thriving developer community. The array of APIs provided by Cognitive Services empowers developers to create applications with advanced voice, vision, and language functionalities, enriching user experiences and propelling innovation [27]. Microsoft's Cognitive Services API serves as evidence of the transformative potential of cloud-based AI services. The platform's diverse APIs and impactful applications underscore the profound influence AI can have across various industries and in daily life. Microsoft's decision to open its technologies to the developer community has not only spurred creativity but has also driven the widespread adoption of AI-powered solutions in today's digital technologies.

2.1.4. IBM Watson Visual Recognition

In early 2015, IBM's Watson brand introduced its Visual Recognition solution, which leveraged technology from Alchemy Vision, a company based in Denver. IBM acquired Alchemy API and used its capabilities to create the Visual Recognition APIs (Figure 2.4). Similar to other leading image recognition providers, IBM offers customized classifier training, allowing developers to tailor the recognition system to their specific needs and applications. This level of customization and integration with IBM's Watson ecosystem makes IBM's Visual Recognition solution a powerful and versatile option for developers seeking advanced image recognition capabilities.

2.1.5. Clarifai

Matthew Zeiler's pioneering work resulted in the establishment of Clarifai, a major image recognition API company. Clarifai provides a variety of services, including picture tagging in over 23 languages and visual similarity searches on previously uploaded photos. Their face-based classifiers are designed to cater to various cultures, and they even provide a photo beauty scorer, focus scorer, and embedding vector builder to assist developers in building their own reverse-image search capabilities.

Clarifai's expertise goes beyond general image recognition, as they have expanded their capabilities to include recognition in specific domains such as weddings, tourism, and hospitality. Their open API provides access to a vast repository of 11,000 picture tagging ideas, making it a valuable resource for developers looking to enhance their applications with advanced image recognition capabilities [9].

With its extensive range of features and domain-specific recognition abilities, Clarifai continues to be a leading player in the image recognition API market, empowering developers to create innovative and tailored solutions for various industries and applications, a simple result shown in Figure 2.5.

Figure 2.4.: Alchemy Vision's analysis delineates the components within the image with notable accuracy. It classifies 'person', 'chair', and 'highchair' as primary classes, recognizes elements of the meal including 'cake mix' and 'food mix', and identifies the face of a young female within the age bracket of 18-24.[55]

2.1.6. Imagga

Imagga has established itself as a notable player in the field of image recognition, offering versatile and powerful solutions in automatic image tagging and categorization. A significant highlight of its capabilities was demonstrated in a study by researchers from Ben-Gurion University, where Imagga was identified as one of the top-performing APIs in a comprehensive comparison of image recognition technologies. This study evaluated several leading APIs, emphasizing Imagga's proficiency in deep learning-based multi-label image classification [48]. Additionally, the Imagga API excels in automating the process of assigning accurate and relevant tags to images, streamlining the categorization and organization of visual content. This feature is particularly beneficial for applications requiring efficient management and analysis of large image sets, as outlined by Meir Fein-

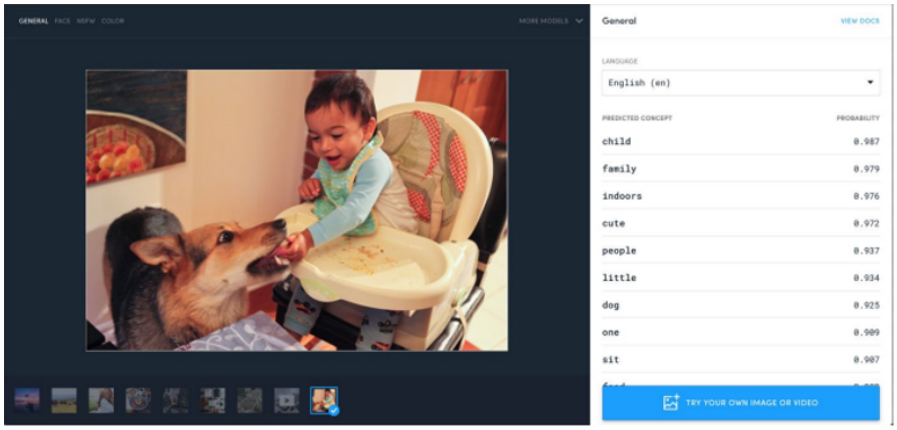

Figure 2.5.: Captured through Clarifai API recognition capabilities, this heartwarming scene is analyzed with keywords reflecting the content and sentiment of the image: a child ('child' - 98.7%) in a family setting ('family' - 97.9%), located indoors ('indoors' - 97.6%). The moment is described as 'cute' (97.2%), with the presence of 'people' (93.7%) and specifically a 'little' (93.4%) one, engaging with a 'dog' (92.5%), with both seated ('sit' - 90.7%). This showcases the API's power to contextually understand and describe a moment.[55]

berg's insights on Imagga's integration with Cloudinary's image management pipeline[36]. These capabilities, coupled with the potential for custom model training, position Imagga as a versatile and robust tool in the realm of computer vision and image analysis.

2.2. Research Significance

In the context of software reuse, it is crucial to evaluate the quality of the software to identify the key factors that contribute to its success. This evaluation involves assessing various aspects of the program, such as its functionality and usability, which are known as quality needs or characteristics [57]. These characteristics may further have sub characteristics, such as learnability under the usability category. To conduct a successful evaluation, it is essential to determine and examine the most critical quality criteria for the given field, using software assessment metrics.

Another important aspect of software evaluation is the consideration of Quality of Service (QoS), which encompasses non-functional qualities such as latency, cost, and correctness. When developers offer functionality as a service, they must choose the most suitable service that aligns with the stated QoS criteria. One approach is to compare available equivalent services and select the one that best meets the requirements [18]. However, research has suggested that combining multiple comparable services can enhance QoS. Despite this potential, previous research has primarily focused on methods to improve accuracy through techniques like majority voting or optimizing other qualities such as dependability, cost, and latency through service composition. Consequently, developers

often lack guidance on how to employ integrated service execution to enhance accuracy, particularly for cognitive tasks.

To address this gap, further research is needed to explore the impact of coupled service execution on overall accuracy. Understanding how combining multiple services can lead to improved accuracy in cognitive activities is essential for developers seeking to optimize their software solutions. This knowledge can inform the development of more effective and efficient software applications that leverage multiple services to deliver higher accuracy and performance.

Additionally, the evaluation of software quality and QoS in the context of service integration and execution can have broader implications for the software industry. By identifying the most critical quality criteria and understanding how to leverage service composition for improved accuracy, developers can make more informed decisions when selecting and integrating various software services. This, in turn, can lead to the development of more reliable, cost-effective, and efficient software solutions that meet the specific needs of users and businesses.

Evaluating the quality of software and considering the impact of service integration and execution on accuracy and QoS are vital steps in the software development process. By assessing various quality characteristics and understanding the potential benefits of combining multiple services, developers can create more effective and high-performance software solutions. Continued research in this area will provide valuable insights and guidance for developers seeking to optimize their software applications, particularly in the domain of cognitive activities. Ultimately, such efforts can lead to significant advancements in the software industry, resulting in better-performing and more reliable software solutions for various use cases.

To measure software quality effectively, software metrics and quality models are essential tools. The ISO 9126 standard, which covers all aspects of software quality, supports both bottom-up and top-down evaluation approaches. Over time, this standard has been revised and updated, and it is now known as ISO/IEC 25010 [18]. The new standard retains the same software quality criteria as before but includes a few adjustments and refinements. The ISO/IEC 25010 standard's eight basic quality characteristics, each with matching sub-characteristics, are depicted in Figure 2.6. These quality characteristics and sub-characteristics provide a complete framework for analyzing and assessing software system overall quality, assisting developers and stakeholders in making educated decisions to improve software performance and user happiness.

Despite the rapid technological advancements in facial recognition systems, the fundamental goal of assessing their performance remains unchanged. An essential aspect of this assessment is the use of operational data in independent evaluations of facial recognition software, which ensures a fair, consistent, and meaningful evaluation of its capabilities.

There are several types of evaluations conducted to assess facial recognition systems. Capability testing aims to determine the feasibility and limitations of a technology, as well as whether it can meet specific application and technical criteria [19]. This type of testing helps in understanding the boundaries and potential of the technology, allowing for informed decision-making in procurement and inspiring the development of new and

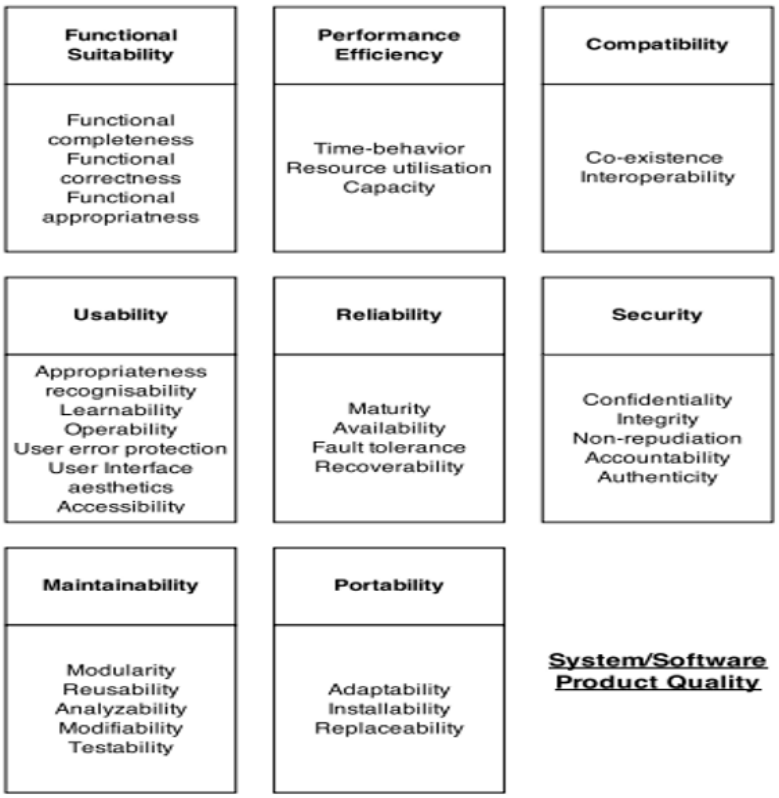

Figure 2.6.: An overview of the ISO/IEC 25010 standard's eight quality characteristics for system and software product quality: Functional Suitability, Performance,..., etc. Each characteristic is broken down into its constituent attributes, such as Functional Completeness, Time-Behavior, and many others, providing a comprehensive framework for evaluating software quality. [51]

improved technologies.

Algorithm comparison testing is another crucial evaluation method that aids in distinguishing which facial recognition technologies perform effectively and which ones fall short. This type of testing is instrumental in guiding procurement choices, as it provides valuable insights into the strengths and weaknesses of various algorithms, helping stakeholders make informed decisions about the most suitable solutions for their specific needs.

Longitudinal performance monitoring is yet another critical aspect of facial recognition evaluation. This involves quantifying the performance improvements of the system over time, which in turn influences upgrade schedules and contract re-competition [26]. Monitoring the performance of facial recognition systems longitudinally ensures that they remain up-to-date and capable of meeting evolving demands and challenges.

The consequences of facial recognition mistakes can be costly. For example, failure to enroll an individual accurately can lead to wasted time, additional procedures, and the need for alternative modalities and processes. As such, the accurate and reliable

performance of facial recognition systems is of paramount importance, especially in applications where security and accuracy are critical.

Assessing facial recognition systems requires various evaluation methods that encompass capability testing, algorithm comparison, and longitudinal performance monitoring. The use of operational data and independent evaluations ensures that facial recognition software is fairly and meaningfully assessed. Accuracy and reliability are key considerations, as mistakes in facial recognition can have significant financial and operational implications. By continuously evaluating and improving these systems, stakeholders can make informed decisions and deploy facial recognition technologies effectively in diverse applications, from security to user service and beyond.

Measuring software metrics and performance is crucial in establishing operational expectations for a program. By characterizing the performance through the use of metrics, developers can identify areas for improvement and implement risk-mitigation measures. For instance, if the enrollment failure rate is high, modifying the environment to limit ambient light could be a suitable mitigation strategy. Relying solely on intuition can be deceptive, as it may not always lead to the most accurate and effective decisions. Having data-driven insights from software metrics provides developers with valuable information to make informed choices and optimize the performance of the system [5].

For example, developers might intuitively believe that using a majority voting technique on the outcomes of multiple equivalent services would enhance accuracy. However, this assumption might not hold true in all cases. By measuring performance and conducting algorithm comparison tests, developers can gain objective insights into the effectiveness of various techniques, helping them make better decisions for optimizing the software's performance and accuracy.

2.3. Recent Developments

Face recognition research has witnessed significant growth and attention in recent times, driven by various factors that have contributed to its prominence. One of the primary catalysts for this surge is the rapid advancement of machine learning methods, which have enabled the development of more sophisticated and accurate face recognition algorithms. Additionally, the availability of robust and free software implementations has made it easier for researchers and developers to experiment and innovate in this field.

The progress in hardware, particularly the availability of faster GPU processors, has also played a vital role in fueling face recognition research. These powerful processors allow for quicker and more efficient computation, making it feasible to process large datasets and complex algorithms at a much faster pace.

The availability of vast sets of web-scraped face photos has been instrumental in training face recognition algorithms. These sets provide a diverse range of facial expressions, poses, and lighting conditions that mimic real-world scenarios, helping researchers create algorithms that are more robust and capable of handling various environmental variations.

Open performance benchmarks and a growing literature on face recognition have further contributed to the field's advancement. These benchmarks provide standardized

evaluations, allowing researchers to compare and assess the performance of different algorithms objectively. The expanding literature and research publications provide insights into the latest advancements and trends, fostering collaboration and knowledge sharing in the face recognition community.

To address the challenges faced in face recognition, new algorithms are continuously being developed to be resistant to variations in posture, lighting, and facial expressions commonly found in photojournalism and social media photographs. Early research on in variance learning relied on a large number of photographs of a few individuals. As the field progressed, larger populations were employed for training, but the benchmarks used remained relatively simple, often involving one-to-one verification with high false match rates.

Over time, researchers have started developing larger-scale identification benchmarks that aim to address the challenges of face recognition at scale. However, the major parameter of these benchmarks often diverges from the high-threshold discriminating challenges required for governmental face recognition applications involving vast populations.

Face recognition research has witnessed significant growth and progress due to factors such as advances in machine learning methods, the availability of powerful hardware, vast datasets, standardized benchmarks, and an increasing body of research. As new challenges emerge, researchers continue to develop more sophisticated and robust algorithms to tackle variations in posture, lighting, and facial expressions. The ongoing efforts in face recognition research are expected to drive further advancements and applications in various domains, including security, surveillance, and user authentication.

Technological advancements in facial recognition have led to a substantial reduction in recognition mistakes. The NIST Face Recognition Vendor Test, one of the most independent and reputable benchmarks, demonstrated that leading commercial algorithms now produce twenty times fewer false negatives on similar portrait photo recognition tasks compared to 2013. This represents a significant improvement compared to the period between 2010 and 2013, where mistake rates only decreased by a factor of two [20].

The improvements in recognition accuracy apply to both high and low false positive identification rates, indicating that Convolutional Neural Network (CNN)-based algorithms can achieve in variance while still maintaining high discrimination capabilities. As a result, major commercial companies have embraced CNN's and are increasingly replacing older methodologies in their facial recognition systems.

The adoption of CNN's in facial recognition has been a game-changer, allowing for more robust and accurate recognition even in challenging conditions, such as varying poses, lighting, and facial expressions. These algorithms can learn complex patterns and features from large datasets, enabling them to generalize unseen data well and improve recognition performance significantly.

The decrease in recognition mistakes has opened up new opportunities for facial recognition technology across various industries. From security and surveillance applications to user authentication and user engagement, CNN-based facial recognition algorithms have become a cornerstone of modern AI-driven systems.

However, despite the significant improvements, challenges related to facial recognition

persist. Issues like privacy and potential misuse of the technology continue to be areas of active research and debate. As facial recognition technology becomes more pervasive, it is essential to strike a balance between leveraging its benefits while safeguarding individual rights and ensuring responsible usage.

Technological advancements, particularly the adoption of CNN-based algorithms, have led to substantial reductions in recognition mistakes in facial recognition systems. The improvements in accuracy across false negative and false positive rates have enabled facial recognition technology to reach new heights in various industries. However, it is crucial to address privacy concerns to ensure responsible and equitable deployment of this powerful technology in the future.

2.4. How Do CNNs Work

Convolutional Neural Networks CNNs are a form of neural network that, when compared to classic feed-forward neural networks, use convolution to minimize the number of weights and simplify calculations. This approach, which is based on the structure of biological neural networks, has been shown to be extremely effective in speech and picture recognition applications.

A typical CNN's architecture consists of numerous layers, including input, convolution, pooling, fully linked, and output layers. The raw data, such as pictures or audio samples, is received by the input layer and processed by the convolutional layers. Filters are used in these levels to extract useful information from the incoming data. The pooling layers serve to minimize computational complexity and avoid over fitting by further reducing the spatial dimensions of the retrieved features.

The fully linked layers come after the pooling layers and are in charge of producing final predictions. They translate the high-level information acquired from the previous layers into relevant outputs based on the current job. In image recognition, for example, fully linked layers may categorize the input picture into distinct categories.

The individual application problem determines the amount of convolution and pooling layers, as well as the type of classifier utilized. The CNN's architecture and settings may be tailored to the task's complexity and needs. This adaptability enables CNNs to be used for a broad range of applications, including image classification, object identification, and natural language processing. Overall, CNNs have transformed the area of artificial intelligence and have become a vital tool for a wide range of applications needing extensive pattern recognition and feature extraction skills.

2.4.1. Convolution in CNN

A 500×500-pixel picture would require 500×500 neurons in the input layer in a typical fully connected neural network. With 108 neurons in the hidden layer, each connection between an input neuron and a hidden neuron would require a weight parameter. The total number of weight parameters between the input and hidden layers is $500 \times 500 \times 108$, for a total of 25×1012 parameters. This huge number of factors might result in a

computationally difficult procedure that necessitates substantial processing resources [21].

A locally connected neural network can be used to alleviate this computational strain and minimize the number of parameters. In this method, filters are used to extract local image elements, then the entire image is interpreted based on these extracted characteristics. For example, if a 10×10 filter is used to examine a local picture region, each hidden layer neuron is linked through the filter to a 10×10 area of the input layer. As a result, there would be 108 filters between the hidden and input layers, each corresponding to a 10×10 region of the input layer. As a consequence, the number of weight parameters between the input and hidden layers is decreased to $10 \times 10 \times 108$, yielding 1010 parameters [21].

By adopting this locally connected approach, the neural network becomes more efficient in terms of computational resources and parameter management. This reduction in parameters not only speeds up the training and inference processes but also allows for the analysis of larger images without excessively increasing computational demands.

The locally connected neural network is particularly beneficial when dealing with high-resolution images or large datasets, whereas the traditional fully connected approach would be impractical due to its high computational cost. This approach has found significant applications in various computer vision tasks, such as image recognition, object detection, and semantic segmentation.

The use of a locally connected neural network can significantly reduce the computational burden and the number of weight parameters compared to a traditional fully connected network. By employing filters to extract local features from an image, this approach allows for efficient analysis and interpretation of large-scale images. As a result, it has become a valuable technique in computer vision and has enabled the development of sophisticated AI systems that can handle complex visual data with greater efficiency and accuracy.

To further reduce the number of weight parameters and computational overhead, the intrinsic properties of natural images can be leveraged. Statistical properties learned from one region of an image can often be applied to other regions as well. By sharing weights and applying the same filter to all parts of an image, the number of weight parameters can be significantly reduced from 25×10^{12} to just 100. This approach considerably lowers the computational effort required for processing images, making it even more efficient [21].

There are two major tasks in the context of face recognition: face identification and face verification. Convolutional Neural Network (CNN) training for these problems generally consists of four steps: face detection, alignment, representation, and classification. The purpose of face identification is to assign an identity to an image from a set of known identities. Face verification, on the other hand, seeks to determine whether a particular image genuinely belongs to a certain individual.

The process begins with face detection, where the CNN identifies and localizes faces within an image. Following this, face alignment is performed to ensure that all detected faces are properly oriented and aligned to a standardized position. Next, the CNN extracts facial features and representations from the aligned faces. These features serve

as a compact and meaningful representation of the facial characteristics that are essential for recognition.

Finally, the classification step takes place, where the CNN uses the extracted features to make decisions. In face identification, the CNN matches the features to a database of known identities to identify the person in the photograph. In face verification, the CNN compares the features of the given photograph with those of a reference image to determine whether they belong to the same individual.

By going through these four steps, the CNN can effectively perform face recognition tasks, providing accurate identification and verification results. Face recognition technology has found wide applications in various industries, from security and law enforcement to mobile devices and social media platforms.

Exploiting the intrinsic properties of natural images allows for a substantial reduction in weight parameters and computational effort. This is particularly beneficial in face recognition tasks, where Convolutional Neural Networks can efficiently handle face identification and verification challenges. By combining face detection, alignment, representation, and classification steps, CNNs can accurately recognize and verify faces, opening up a wide array of practical applications in today's technologically advanced world.

Deep learning algorithms, particularly Convolutional Neural Networks CNNs, have made significant advancements in improving face recognition and verification. These algorithms have proven to be highly effective in extracting meaningful facial features and representations from images. Additionally, efficient loss functions have been developed to train CNNs effectively, favoring representations with low intra-class variability (within the same identity) and high inter-class variability (between different identities). These loss functions enhance the discriminative power of the learned face representations, leading to more accurate and compact feature embeddings [2].

However, successful face recognition procedures rely not only on advanced CNN architectures and loss functions but also on the initial steps of face detection and fiducial landmark/key-point localization. Face detection algorithms are essential for locating and localizing faces within an image, providing the necessary region of interest for subsequent recognition tasks. Accurate face detection greatly influences the overall recognition performance.

Similarly, fiducial landmark/key-point localization algorithms play a crucial role in face recognition, as they identify key points on the face, such as the eyes, nose, and mouth. These points serve as reference landmarks for face alignment, which ensures that all detected faces are correctly oriented and aligned to a standardized position. Proper alignment is essential for achieving consistent and robust feature extraction from the faces.

Recent advances in the use of large-scale datasets and CNNs have significantly improved face detection and landmark localization performance. The availability of extensive annotated sets has allowed CNNs to learn intricate patterns and spatial relationships, resulting in more accurate and reliable face localization.

Face recognition technology has advanced dramatically as a result of the combination of deep learning algorithms, CNN structures, efficient loss functions, and enhanced face

detection and landmark localization approaches. These advancements have prepared the way for widespread use of facial recognition systems in a variety of fields such as security, surveillance, biometric identification, and human-computer interaction (Figure 2.4).

2.5. Other Deep Learning Networks

Deep neural networks are deep learning architectures with numerous hidden layers between the input and output layers. These networks support both forward and backward learning, making the training process easier. Deep learning architectures exist in a variety of flavors, and many of them stray from the norm. These designs may be classified into three forms based on connectionist models of neurons, as shown in Figure 2.7 completely connected, locally connected, and several other deep learning networks. Each architecture has its own set of strengths and capabilities, making it suited for a variety of activities and datasets. The architecture chosen is determined by the unique task at hand as well as the required performance characteristics. Deep learning has transformed industries ranging from computer vision and natural language processing to speech recognition and autonomous systems, propelling significant advances in artificial intelligence and machine learning.

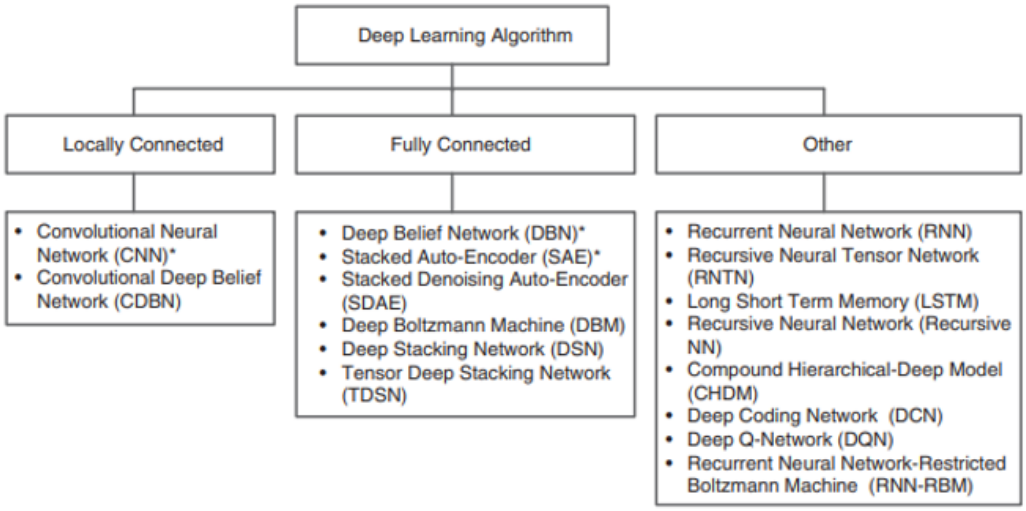

Figure 2.7.: Exploring the intricacies of neural architectures, this diagram classifies deep learning algorithms into three main categories: Locally Connected, Fully Connected Networks and Other Networks, which encompass Recurrent Neural Networks (RNNs) and their variants. Each category is crucial for understanding the diverse approaches and structures used in the field of deep learning [25]

2.5.1. Connectivity of Deep Learning Networks

Based on connectionist models of neurons, deep learning architectures may be roughly categorized into three types: totally connected, locally connected, and other deep learning networks. Deep Belief Networks DBN and Deep Boltzmann Machines DBM are fully linked networks that include links from the input to the output layer, allowing information to be sent across all levels. Locally linked networks, such as Convolutional Neural Networks CNN, on the other hand, use local feature connections between the input and output layers. This solution makes use of partial connections and weight sharing to decrease the amount of weights while efficiently capturing local properties [23].

Other deep learning architectures that fulfill specialized goals exist in addition to completely and locally linked networks. Architectures built for sequential data processing, such as time series or natural language data, include Recurrent Neural Networks RNN and Long Short-Term Memory LSTM networks. Because RNNs feature feedback links that allow information to remain over time, they are well-suited for sequential tasks. LSTMs, a kind of RNN, can capture long-term relationships and are especially effective for dealing with extended data sequences.

However, it's essential to note that while RNNs and LSTMs are powerful for certain tasks, they may face challenges when dealing with data streams that require real-time processing. The sequential nature of their computations can lead to computational bottlenecks and delays, making them less effective for real-time applications or scenarios with high-speed data streams.

Each sort of deep learning architecture has advantages and disadvantages that make it ideal for various applications. The architecture used is determined by the kind of data, the complexity of the task, and the required performance characteristics. Deep learning has shown amazing capabilities in a variety of fields, including computer vision, natural language processing, speech recognition, and robotics, resulting in significant advances in artificial intelligence and machine learning. Expect increasingly more complex and specialized architectures to develop as research in these subject advances, substantially broadening the potential of deep learning applications.

2.5.2. Recurrent Neural Networks RNNs

In contrast to a typical Artificial Neural Network ANN with three layers at each time point, a Recurrent Neural Network RNN has a unique characteristic. In addition to considering the current input, an RNN also takes into account the prior outputs from the data stream. This means that information processing at each time step takes into account not only the current input but also the result created in the preceding time step. RNNs can capture sequential dependencies and temporal patterns in data because of their recurrent nature, making them well-suited for applications requiring time series, natural language processing, and sequential data analysis. RNNs' capacity to remember prior knowledge offers them a significant advantage when dealing with dynamic and sequential data, allowing them to make predictions and judgments based on historical context. As a result, RNNs have found significant applications in fields like speech recognition,

language translation, sentiment analysis, and stock market prediction, where sequential data processing is critical for achieving accurate and meaningful results.

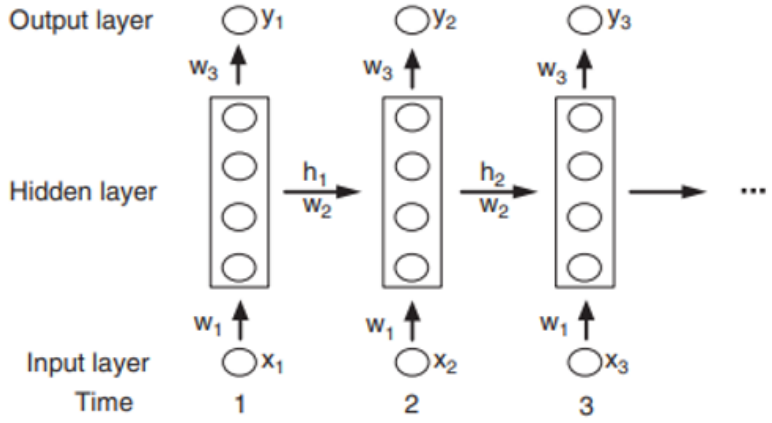

Figure 2.8.: A schematic representation of a Recurrent Neural Network (RNN) layer, illustrating the flow of data through time. Each panel shows the progression of the input layer, through a hidden layer, and onto the output layer at successive time steps, highlighting the network's ability to pass information along the timeline, a fundamental feature of RNNs used for processing sequences in tasks such as language modeling... [25]

When dealing with data sequences spanning numerous time steps, training a Recurrent Neural Network RNN is equivalent to training a feed-forward network with hundreds of layers. When processing a sequence of data, unlike feed-forward neural networks, which process information in a one-way flow from input to output, an RNN must remember its prior output in order to compute the current output again. This is accomplished by connecting nodes in the network structure's hidden layers [2]. RNNs take in sequences of values and output sequences of values, making them ideal for jobs involving temporal dependencies, such as language and voice recognition algorithms.

At each time step t, an RNN can be conceptualized as a three-layer Artificial Neural Network ANN. At time t, the input is designated as xt, the output is denoted as yt', and the hidden layer is denoted as ht. At all time steps, the same set of network parameters, such as connection weights, is utilized. $w1$, $w2$, and $w3$ indicate the connections between the input and hidden layers, the hidden layer at time $t-1$, and the hidden layer and output layer, respectively. The hidden layer value ht is calculated by taking the current input value xt and the hidden layer value from the previous time step, $ht-1$, into account. Because of this recursive process, the RNN can replicate temporal activity and capture sequential patterns, making it ideal for evaluating time-series data and sequences.

RNNs may be employed in a range of practical applications, including language modeling, sentiment analysis, handwriting recognition, and stock price prediction. They excel in jobs requiring sequential data comprehension and processing, where context and past knowledge are critical for accurate forecasts and decision-making.

However, due to difficulties such as disappearing or ballooning gradients, which can hamper learning and result in poor performance, training RNNs can be difficult. To solve these issues, numerous modifications and enhancements have been proposed, such as Long Short-Term Memory LSTM networks and Gated Recurrent Units GRU, which are intended to alleviate the vanishing gradient problem and increase RNNs capacity to preserve long-term dependencies.

RNNs are essential in deep learning and have been shown to be effective tools for processing sequential data. Their capacity to capture temporal dynamics and sequential patterns makes them an invaluable tool in a variety of applications, including as natural language processing, audio recognition, and time-series forecasting. As study in this field continues, may expect more improvements and developments.

2.5.3. Recursive Neural Tensor Networks RNTN

The Recursive Neural Tensor Network (RNTN) is a recursive deep neural network that can handle input data of varying lengths and support multistage predictions [24]. While both RNTN and RNN exhibit recursive behavior, they do so in different ways. RNN employs recursion based on time sequencing, whereas RNTN employs recursion based on data structures known as tensors. This distinct approach allows RNTN to effectively process sequences and hierarchical structures, making it a valuable tool for tasks that involve complex and nested data patterns.

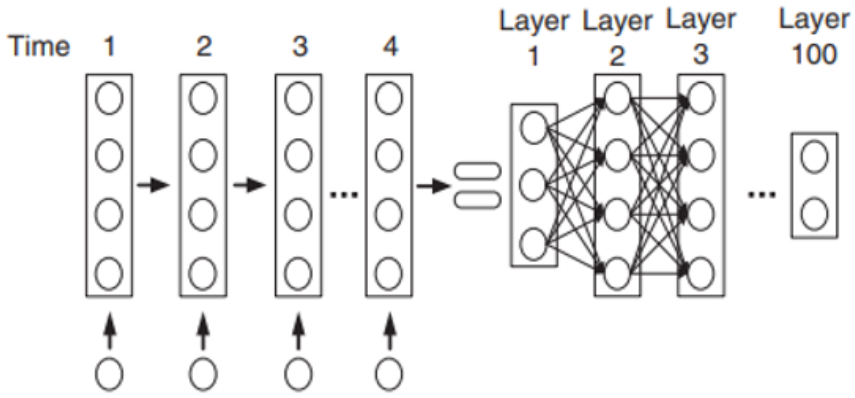

Figure 2.9.: Depicting the complexity of a Recursive Neural Tensor Network (RNTN), this diagram illustrates the sequential data flow across time-steps before entering a densely connected network of layers. This network structure is known for its deep learning capabilities, handling hierarchical data representations, and is commonly applied to natural language processing tasks.[51]

2.5.4. Input and Output Relationships in Different Neural Networks

Because the Recurrent Neural Network RNN can handle both input and output sequences, it is suitable for a wide range of applications. Figure 2.10 depicts four potential situations

for input-output pairs in various jobs.

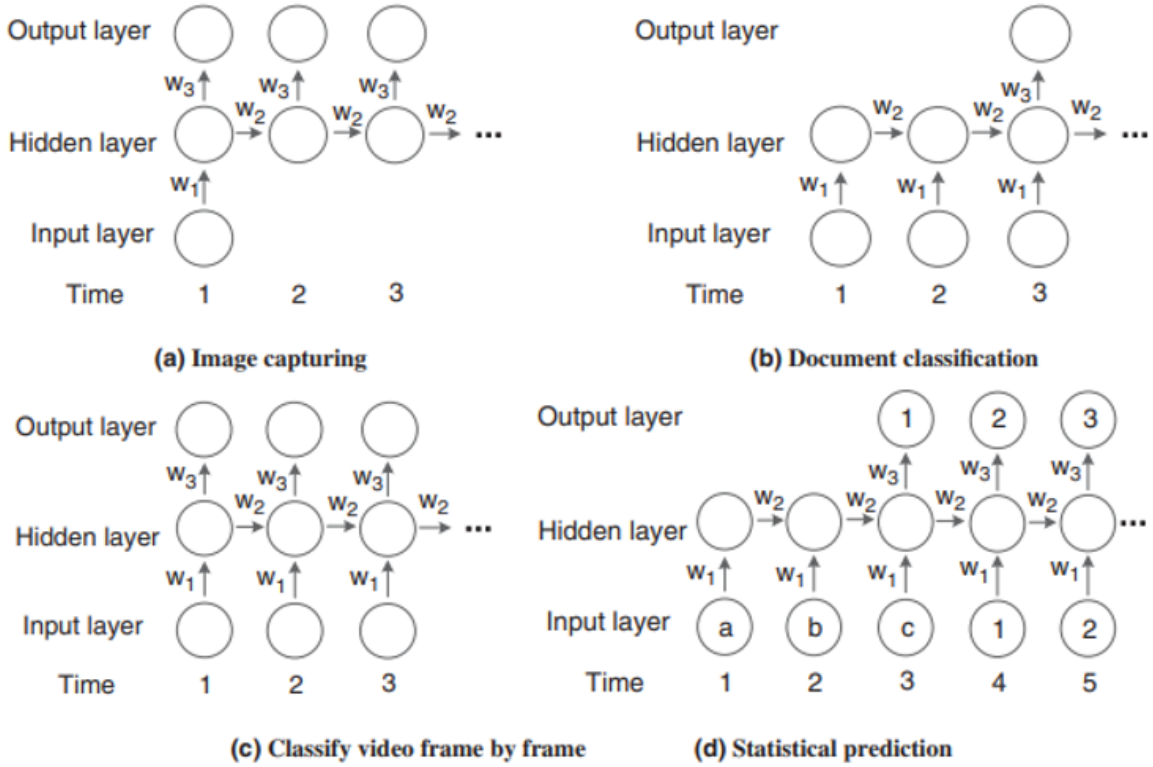

Figure 2.10.: An illustrative guide to Recurrent Neural Network (RNN) configurations for various applications: (a) shows an RNN used for image capturing over time, (b) demonstrates document classification through sequential processing, (c) depicts frame-by-frame video classification, and (d) represents an RNN designed for statistical prediction, highlighting the adaptability of RNNs to different data types and temporal processes.[25]

Figure 2.10(a) depicts an input-output configuration that is ideal for picture captioning applications. In this case, the RNN receives an image as input and provides a descriptive caption as output.

Figure 2.10(b) depicts an input structure with multiple inputs and a single output, which is commonly used in document categorization tasks. The RNN processes multiple pieces of text data as inputs and predicts a single category or label as output.

Figure 2.10(c) demonstrates both the input and output as sequences, a configuration often used in RNNs for video streaming applications, where the network processes frames of a video one by one to generate a sequence of predictions. This design is also suitable for statistical forecasting tasks, where the RNN predicts future events based on past sequences of data.

Figure 2.10(d) presents a scenario where known data is provided at times 1 and 2, and the RNN starts making predictions from time 3 onward. At time 3, when data is input, the output result is 1, indicating the forecast for the next time step. Similarly, inputting data 1 at time 4 results in predicting data at time 5 as 2, leading to an output result of 2.

Because of RNN's capacity to handle a variety of input-output combinations, it is a valuable tool for jobs involving sequential data and time-series analysis. Its recurrent nature helps it to remember previous knowledge, allowing it to make predictions and judgments based on historical context. This trait is especially useful for applications requiring the interpretation and processing of data sequences, such as natural language processing, speech recognition, and time-series forecasting.

Novel versions and architectures are being created as RNN research continues to solve difficulties and improve performance in certain jobs. To increase RNNs' capacity to capture long-term dependencies and reduce difficulties such as disappearing or ballooning gradients during training, modifications such as Long Short-Term Memory (LSTM) networks and Gated Recurrent Units GRU have been proposed. These developments continue to broaden RNN capabilities and push the boundaries of what is possible in sequential data processing.

2.6. Other Deep Learning Neural Networks

Convolutional Deep Belief Networks CDBN are a fusion of Convolutional Neural Networks CNN and Deep Belief Networks DBN to address the challenges of scaling DBN to handle large, high-dimensional images effectively.

Deep Q-Networks DQN, on the other hand, are deep neural network structures developed by Google DeepMind that combine the Q-learning reinforcement learning approach with artificial neural networks [26]. DBMs, or Deep Boltzmann Machines, consist of one visible cell layer and multiple hidden cell layers, with no connections within the same layer. DBM is a deep architecture formed by stacking several Restricted Boltzmann Machines RBM together, with undirected connections between any two levels.

The integration of different neural network architectures and learning algorithms allows for the development of powerful models that can tackle complex tasks. CDBNs leverage the strengths of both CNN and DBN, enabling effective image recognition and feature extraction, while DQNs combine reinforcement learning with deep learning to achieve impressive results in challenging decision-making problems. These innovations have contributed significantly to the advancement of AI and machine learning, enabling the development of sophisticated systems capable of processing and understanding complex data, such as images, videos, and natural language. As research in this field continues, expect further advancements in deep learning techniques and their applications, leading to more robust and intelligent AI systems.

Stacked Denoising Auto-Encoder SDAE is a variation of stacked auto-encoders that utilizes Denoising Auto-Encoders DAE instead of regular auto-encoders AE. DAE employs unsupervised training, involving three steps: data corruption, encoding, and decoding [54]. Deep Stacking Network DSN constructs deep networks by combining basic neural

network modules, where each module's output represents a category, and the input for subsequent layers is a concatenation of the original data (x) with the output (y) of the previous layers [26].

Tensor Deep Stacking Network TDSN is an extension of Deep Stacking Networks DSN that incorporates multiple stacked blocks. Each block contains three layers: an input layer (x), two parallel hidden layers ($h1, h2$), and an output layer (y) [31]. Long Short-Term Memory LSTM is a step forward from Recurrent Neural Networks RNN that incorporates memory modules in the hidden layers. By considering information from previous time points, LSTM effectively overcomes the challenges of vanishing and exploding gradients encountered in training simple RNNs.

Deep Coding Networks DPCN are hierarchical generative models that represent a deep learning network capable of self-updating by utilizing contextual data [57]. Compound Hierarchical-Deep Model CHDM integrates deep networks with non-parametric Bayesian models. In CHDM, deep architectures like Deep Belief Networks DBN, Deep Boltzmann Machines DBM, and stacked autoencoders are employed for learning features [13]. A recurrent temporal Restricted Boltzmann Machine RNN-RBM combines recurrent neural networks and restricted Boltzmann machines to capture temporal dependencies in sequential data.

These diverse deep learning architectures highlight the continuous efforts to improve the performance and capabilities of deep neural networks. By combining different techniques and models, researchers aim to address specific challenges in various applications, from image recognition and natural language processing to time-series analysis and reinforcement learning [75]. These advancements in deep learning have fuelled the rapid progress of artificial intelligence, enabling the development of sophisticated systems with the ability to understand and process complex data, making significant contributions to various domains and industries. As research in this field continues, expect further innovations in deep learning architectures, leading to even more powerful and versatile AI systems in the future.

2.7. Datasets

Deep convolutional neural networks CNNs rely on vast amounts of training data, and large corporations have access to private image collections with hundreds of millions of photos. These sets are utilized to train large-scale face recognition networks [23]. While publicly available face recognition databases continue to improve in size and quality, they are still smaller in comparison to corporate datasets. Face recognition involves three fundamental stages: face detection, landmark localization for alignment, and face classification, and each of these phases requires a specific set of data [31]. Access to these extensive and diverse sets enables companies to train highly accurate and robust face recognition systems, leading to advancements in various applications, from security and surveillance to user authentication and personalized services. However, the ethical use and handling of such large sets remain important concerns in the development and deployment of face recognition technology.

The introduction of deep CNNs led to the creation of larger training datasets like CelebA, CASIA-WebFace, UMDFaces, MS-Celeb-1M, and VGGFace [24]. However, these sets have limitations as they mainly consist of static photographs of celebrities taken under controlled lighting conditions. On the other hand, assessment datasets often include videos and images captured under challenging and varied conditions, leading to a domain shift between the training and assessment sets. To address this issue, researchers have proposed video sets such as YouTube Faces YTF and UMDFaces Video, which provide more diverse and robust training data. Combining still image and video sets during training has been shown to improve performance as the learned features are more robust and generalize better to real-world scenarios [9].

The availability of such extensive datasets has significantly contributed to the advancement of face recognition technology. Deep CNNs trained on these sets have demonstrated remarkable accuracy in recognizing faces even in challenging conditions, such as variations in lighting, facial expressions, and occlusions. Furthermore, the use of transfer learning, where a pre-trained model is fine-tuned on a specific task, has become a common practice, allowing researchers and developers to leverage the knowledge and features learned from large-scale sets to boost performance on smaller or specialized sets. Despite the progress, face recognition technology still faces challenges, including issues related to privacy, bias, and fairness. The use of facial recognition in surveillance and security applications has raised concerns about potential misuse and infringement of individual privacy rights. Additionally, biases present in the training data can result in algorithmic discrimination, where certain groups may be misidentified or underrepresented [75].

The availability of large and diverse datasets has played a crucial role in the advancement of face recognition technology. One of the most prominent sets used for training and evaluation is the WIDER FACE set, which contains approximately 400,000 annotated faces. Prior to the WIDER FACE set, the FDDB set was commonly used, but its size and variability were limited, necessitating the need for larger and more complex sets to effectively train and evaluate modern face identification techniques [67]. The IARPA JANUS Benchmark datasets, such as IJB-C, have also contributed significantly to the development of face recognition algorithms. IJB-C contains around 149,000 still images and video frames, providing a vast number of face annotations for evaluating face recognition and identification in unconstrained and real-world situations [10]. These sets include a wide range of variations in lighting conditions, facial expressions, occlusions, and poses, making them valuable for testing the robustness and generalizability of face recognition models.

For facial keypoint recognition, datasets like AFW, 300 faces-in-the-wild, LFPW, and the largest, AFLW, with 26,000 annotated faces, are widely used. These sets focus on detecting and localizing facial keypoints, such as eyes, nose, and mouth, which are essential for tasks like face alignment and expression analysis [18].

However, it is crucial to acknowledge that not all datasets are perfect, and some may have limitations or inaccuracies in their annotations. For instance, some sets, like UMDFaces and UMDFaces Videos, rely on automatic annotation methods using pre-trained models, which may introduce errors or inconsistencies in the data [10]. Such

imperfections can impact the training of deep neural networks and affect the overall performance of face recognition systems.

Researchers and developers must be cautious when selecting and using datasets for training and evaluation purposes. Ensuring the accuracy and representativeness of the datasets is essential to obtain reliable results and develop robust face recognition algorithms. Furthermore, addressing biases and ensuring fairness in dataset creation and usage is critical to prevent algorithmic discrimination and promote responsible deployment of face recognition technology in various applications. Table: 2.1 below provides different datasets by their type and sizes.

2.7.1. Bias in Datasets

One such type of bias is "sampling bias," which occurs when the training data used for a machine learning model is not representative of the actual population it is meant to generalize to. For instance, if the training data primarily includes data from a specific demographic group, the model may not perform well when applied to individuals from other demographics, leading to biased predictions.

Another form of bias is "labeling bias," which arises when the labels assigned to the data during the training process contain subjective judgments or cultural prejudices. For example, if a dataset is labeled in a way that reflects certain stereotypes or prejudices, the model may learn and perpetuate those biases during inference [67].

Moreover, "historical bias" refers to the presence of past discrimination or unequal treatment in the data that can influence the model's predictions. If historical inequalities are present in the data, the model may learn to perpetuate these disparities, leading to biased outcomes.

"Representation bias" is another critical concern, which arises when the data used for training does not adequately represent all possible variations or classes in the real world. As a result, the model may struggle to generalize accurately to unseen or underrepresented instances, leading to biased results.

Addressing biases in machine learning models is crucial to ensure fairness and equitable outcomes. One approach to mitigate biases is through data preprocessing, where techniques like data augmentation and oversampling can be used to balance the representation of different classes in the training data [83]. Additionally, careful examination and selection of features and labels can help reduce the influence of subjective or prejudiced judgments in the training process.

Furthermore, the development of diverse and inclusive teams in AI research and development can also help identify and address biases in the data and algorithms [15]. By incorporating diverse perspectives and experiences, AI systems can be more robust and less likely to perpetuate discriminatory outcomes.

Understanding and addressing biases in machine learning models are essential for the responsible and ethical deployment of AI systems, particularly in applications like face recognition, which can have significant social and ethical implications. By acknowledging and actively working to reduce biases, ensure that AI technologies contribute positively to society and do not perpetuate or exacerbate existing inequalities.

2.7.2. Selection bias

Another type of bias that can affect machine learning models is "confirmation bias," which occurs when the training data reinforces existing stereotypes or preconceptions. If the training data contains examples that are biased towards certain beliefs or opinions, the model may learn to favor these biases and make predictions that align with those beliefs.

Furthermore, "group bias" can emerge when the data used for training contains unequal representation of different demographic or social groups [44]. This can lead to the model making biased predictions that favor or disfavor certain groups, resulting in unfair outcomes.

Addressing bias in machine learning models is a complex and ongoing challenge. It requires a combination of careful data collection and preprocessing, algorithmic fairness techniques, and continuous monitoring and evaluation of model performance [57]. Transparency and explainability of AI systems are also crucial to understanding and identifying potential biases in the decision-making process.

To combat biases, efforts are being made to develop standardized guidelines and benchmarks for evaluating fairness in machine learning models. Researchers and practitioners are exploring methods to reduce bias in training data, such as adversarial training and fairness-aware learning [65]. Additionally, incorporating ethical considerations and diverse perspectives into the design and deployment of AI systems can help mitigate biases and promote equitable outcomes.

Biases in machine learning models can have significant consequences, especially in applications like language translation and face recognition [2]. As AI technologies become more pervasive in daily lives, it is essential to recognize and address biases to ensure fairness, inclusivity of AI systems. By adopting a holistic and multidisciplinary approach, allows work towards creating AI technologies that benefit all individuals and contribute positively to society (Figure 2.11).

Figure 2.11.: Exploring Gender Neutrality in Translation: This image captures Google Translates interpretation of English sentences into Turkish and back, highlighting the gender-neutral term 'o' in Turkish which stands for 'he' or 'she', reflecting the language's approach to gender in pronouns.[55]

2.7.3. Implicit bias

Texture bias is a type of bias that occurs when machine learning models rely too heavily on the texture or appearance of an object to make predictions, rather than considering

the context or surrounding information. This bias can lead to misinterpretations and incorrect classifications, as the model may prioritize certain visual features over others [52]. In the example provided, a texture-biased image recognition model may mistakenly classify an image of a sofa with zebra-striped fabric as a zebra, simply because it focuses on the texture of the stripes rather than recognizing the overall context of the image.

To address texture bias and improve the generalization of machine learning models, researchers are exploring techniques such as data augmentation, which involves adding variations to the training data to expose the model to different contexts [24]. Additionally, using more diverse and representative datasets can help reduce bias and improve the robustness of the model's predictions. As AI technologies continue to advance, it is crucial to be aware of the potential biases that can arise in machine learning models and take proactive measures to mitigate them. By understanding the various types of biases and continuously improving the algorithms and data used in AI systems, allow creation more accurate, fair, and reliable AI applications [76] as in Figure 2.12.

Figure 2.12.: Optical Illusion in Upholstery: This image features a loveseat upholstered in a black and white zebra stripe pattern, creating a striking visual effect that could easily be mistaken for the hide of a zebra in a monochromatic setting.[55, 66]

Data Type	Name	Details
Image	Open Images V4	Nine million images in 19,700 categories.
		1.74 million images with 600 categories (bounding boxes).
Image	Microsoft COCO	330,000 images with 80 object categories.
		Contains bounding boxes, segmentation, and five captions per image.
Video	YouTube-8M	6.1 million videos, 3,862 classes, 2.6 billion audio-visual features.
		3.0 labels/video.
		1.53 TB of randomly sampled videos.
Video, images	BDD100K (from UC Berkeley)	100,000 driving videos over 1,100 hours.
		100,000 images with bounding boxes for 10 categories.
		100,000 images with lane markings.
		100,000 images with drivable-area segmentation.
		10,000 images with pixel-level instance segmentation.
Video	Waymo Open Dataset	Approximately 25 million 3D bounding boxes.
		22 million 2D bounding boxes.
Text	SQuAD	150,000 Question and Answer snippets from Wikipedia.
Text	Yelp Reviews	Five million Yelp reviews.
Satellite	Landsat Data	Several million satellite images (100 nautical mile width and height).
		Eight spectral bands (15- to 60-meter spatial resolution).
Audio	Google AudioSet	2,084,320 10-second sound clips from YouTube with 632 categories.
Audio	LibriSpeech	1,000 hours of read English speech.

Table 2.1.: Comparison of different datasets by their type and sizes. [10]

2.8. Deep Learning Advances and Challenges

Significant strides have been achieved in the realm of deep learning technology, particularly in the context of image categorization and identification. Deep neural networks, specifically convolutional neural networks (CNNs), have demonstrated exceptional performance in tasks such as image classification, object localization, and object identification in films [83]. The prowess of CNNs, highlighted in competitions like ImageNet, has propelled the field of deep learning in computer vision

One intriguing aspect of CNNs is their application in style transfer, a process that involves transforming the content of one image to emulate the artistic style of another [24]. Researchers have achieved impressive results in image transformations by approximating activations at specific layers of the original content and utilizing the Gram matrix of activation from the style image. Fine-tuning the final output is possible by adjusting hyper parameters in the loss function and selecting appropriate layers for optimization.

In the domain of natural language processing (NLP), deep learning has made remarkable progress, benefiting language translation, speech-to-text recognition, and sentiment analysis through recurrent neural networks (RNNs) and long short-term memory (LSTM) networks. The combination of CNNs with RNNs has led to the development of generative models capable of describing images provided as input [52].

The continual evolution of deeper models, advancements in computing power, and access to larger datasets have played pivotal roles in the success of various deep learning architectures. This progress has not only facilitated the expansion of deep learning technologies into other domains but has also impacted video applications [2]. For example, video datasets have been instrumental in training deep neural networks for tasks like action recognition, video summarization, and scene understanding.

Despite the remarkable achievements, challenges persist in the field of deep learning, particularly in terms of interpretability. The intricate nature of deep neural networks makes it challenging to comprehend the rationale behind specific decisions. Ongoing initiatives aim to develop techniques that explain and interpret the decisions made by these models, crucial for ensuring transparency and trust in AI applications [53].

Deep learning has undeniably revolutionized fields like computer vision and natural language processing. CNNs have excelled in image-related tasks, while RNNs and LSTMs have significantly advanced language processing applications [2]. As these technologies evolve, their potential for solving intricate problems and transforming various industries remains exciting and promising. However, addressing challenges in interpretability and other considerations is essential to ensure the responsible and beneficial use of AI technologies

The impact of DeepMind's AlphaGo2 project on artificial intelligence was monumental, especially in its victory over the renowned Go player Lee Sedol. The complex nature of Go, with its expansive search space, presented a formidable challenge for AI [9]. AlphaGo2 utilized innovative techniques, employing CNNs to describe positions on the Go board. These networks were trained through supervised learning, utilizing human expert plays to develop a policy network for intelligent moves based on the board situation [13].In

the broader context of AI recognition, AlphaGo2's success serves as a compelling case study, showcasing the potential of deep learning in tackling complex tasks and pushing the boundaries of artificial intelligence. Reinforcement learning played a pivotal role in optimizing the policy network through self-play, progressively enhancing its strategies over time. Value networks were seamlessly integrated to evaluate board situations and predict potential game outcomes.

A critical element contributing to AlphaGo2's success was the adoption of the Monte Carlo tree search method, enabling AI to assess the value of each state within the vast tree of all possible move sequences for informed and strategic decision-making [34]. Asynchronous multi-threaded searches on CPUs managed the computational demands, while parallel calculations on GPUs significantly enhanced system efficiency. The final iteration of AlphaGo2 featured 40 search threads, 48 CPUs, and 8 GPUs, with a distributed version undergoing testing with an impressive configuration of 40 search threads, 1,202 CPUs, and 176 GPUs. During the pivotal match with Lee Sedol, a specialized ASIC known as the Tensor Processing Unit (TPU) was employed, renowned for its energy efficiency and optimization for machine learning tasks, rendering it ideal for complex AI applications like AlphaGo2 [9].

AlphaGo2's triumph over Lee Sedol marked a watershed moment in artificial intelligence, exemplifying the potential of deep learning techniques and sophisticated algorithms in addressing intricate strategic tasks. This achievement has paved the way for similar AI technologies to be deployed across diverse domains such as natural language processing, computer vision, and robotics, unlocking avenues for addressing real-world challenges and propelling AI research forward.

Beyond the automotive and surveillance sectors, computer vision technology has witnessed extensive applications across various industries [67]. In the medical field, computer vision algorithms assist in analyzing medical images, aiding doctors in diagnosing and treating diseases. For instance, in radiology, deep learning algorithms can detect abnormalities in X-rays and MRI scans, facilitating more accurate and timely diagnoses.

In the retail industry, computer vision serves multifaceted purposes, including inventory management, user analytics, and augmented reality shopping experiences. Smart shelves equipped with cameras automatically monitor inventory levels, alerting store managers when restocking is necessary. Moreover, computer vision plays a pivotal role in analyzing user behavior and preferences, enabling retailers to optimize store layouts and marketing strategies [53].

The agricultural sector harnesses computer vision for precision farming practices, where drones outfitted with cameras capture aerial images of crops. This enables farmers to monitor plant health and identify areas requiring attention, resulting in more targeted and efficient resource utilization, increased crop yields, and reduced environmental impact [77]. This comparison underscores the transformative impact of computer vision across industries, highlighting its role in enhancing efficiency, productivity, and sustainability through advanced AI recognition technologies.

Computer vision is reshaping the entertainment and gaming industries, primarily through the integration of virtual reality (VR) and augmented reality (AR) applications,

which create immersive and interactive experiences. AR games, for instance, enhance the gaming experience by overlaying digital content onto the real-world environment.

In the realm of security, computer vision technologies play a pivotal role in biometric identification and access control, ensuring secure authentication and restricted area access through facial recognition systems. Additionally, computer vision aids in detecting suspicious behavior and potential security threats in public spaces [54].

The rapid progression of computer vision technology has ushered in an era of automation, efficiency, and innovation across various sectors. As researchers refine and develop these technologies, the applications are virtually limitless, spanning healthcare, agriculture, entertainment, and security [57]. Computer vision extends beyond facial authentication and image recognition; it delves into detecting diseased regions inside the human body through medical imaging techniques like computed tomography (CT) and magnetic resonance imaging (MRI), promising early detection and improved treatment outcomes for various medical conditions [9].

The integration of computer vision into medical imaging has unveiled new avenues for early disease diagnosis. Advanced algorithms scrutinize medical scans, identifying abnormal patterns or growths indicative of underlying health issues. By enabling early disease detection, this technology holds the potential to save numerous lives and alleviate the burden on healthcare systems [22]. Another significant application of computer vision is in generating 3D images of living cells. A device has been developed that can produce high-resolution 3D images of cells within minutes. This breakthrough technology empowers scientists to explore pharmaceutical effects at the single-cell level, leading to a deeper understanding of drug interactions and potential disease treatments.

In the entertainment industry, "matchmove" technology plays a crucial role in producing visual effects for movies and television shows. In this process, actors perform in a blue sheet studio, and computer-generated backdrops are added later based on camera perspective adjustments and 3D information [13].The outcome is a seamless integration of real-life performances with digitally created environments, delivering stunning and immersive visual experiences for audiences [9]. Additionally, "projection mapping" showcases the versatility of computer vision in creating captivating interactive experiences. By projecting images onto three-dimensional structures like buildings, artists and designers can transform static objects into dynamic displays. The system analyzes the 3D data of the object, adjusting the projected images accordingly, enabling visuals to adapt to surface transitions and even respond to movements in the surrounding area [5]. The comparison analysis of AI recognition underscores the transformative impact of computer vision across diverse industries, highlighting its role in enhancing efficiency, innovation, and user experiences.

As computer vision technology continues to evolve, its potential applications appear limitless. Industries across healthcare, entertainment, architecture, and marketing are embracing this technology to redefine their operations. Moreover, the integration of computer vision with other cutting-edge technologies like augmented reality is unlocking even more thrilling prospects for the future [76]. Computer vision is making significant strides in various domains, positively impacting lives in diverse ways. From advancing medical imaging and pharmaceutical research to crafting captivating visual effects and

immersive experiences, the influence of computer vision continues to expand, promising a future brimming with boundless opportunities for innovation and advancement [53]. As researchers and technologists push the boundaries of this field, expect witnessing more remarkable advancements that will shape world for the better.

Gesture recognition technology has emerged as a prevalent user interface, reshaping the dynamics of human-computer interaction. By enabling users to execute actions through gestures and voice commands, gesture recognition has eliminated the need for physical touch, offering a more intuitive and hygienic experience [24]. Its applications span various sectors, with notable impacts observed in the medical field.

In medicine, gesture recognition has significantly impacted surgical procedures, empowering surgeons to manipulate computers and access information during surgery without direct physical contact. Surgeons can review critical data like X-ray images without touching the computer screen, maintaining sterility in the operating room [13]. Additionally, medical staff can control computers through gestures or voice instructions, facilitating seamless communication and access to vital information during critical moments.

Simultaneously, constructing image understanding systems for knowledge-based vision systems poses unique challenges. The primary hurdle lies in aligning the representation of the target model with the cues present in the observed image [17]. Analogous to knowledge systems where information representation influences the inference process, the representation of target models in vision systems plays a pivotal role in the matching process.

Traditionally, 3D vision systems have represented models separately, with various proposed methods to achieve this. However, dealing with complex objects poses challenges in differentiating between the object and the model if they remain distinct entities. To address this, it's crucial to establish linkages between individual models within the representation, enabling the computer to develop structural knowledge [20, 24, 50]. This interconnected model representation enhances the system's ability to accurately understand complex scenes and objects.

Recent research and development efforts have focused on enhancing the efficiency and accuracy of these interconnected vision systems. By integrating machine learning techniques and advanced algorithms, vision systems can recognize patterns and relationships within the data more effectively. This has opened up new possibilities in fields like robotics, autonomous vehicles, and augmented reality, where a deep understanding of the environment is crucial for intelligent decision-making [83].

Moreover, the integration of gesture recognition with knowledge-based vision systems holds tremendous potential. Envision a future where surgeons can navigate medical images and patient records using simple hand gestures or voice commands during surgery, without losing focus [53]. This seamless interaction could potentially lead to improved surgical outcomes and enhanced patient care, showcasing the transformative power of AI recognition in healthcare and beyond.

As technology advances, the integration of gesture recognition and vision systems is poised to revolutionize various industries [24]. From healthcare and manufacturing to entertainment and education, the prospect of interacting with computers and devices

through intuitive gestures and voice commands heralds a more efficient and immersive future. With researchers and engineers continually pushing the boundaries of these technologies, the potential for innovation and enhancement seems boundless [57]. The journey toward a world of interconnected gesture-controlled vision systems is well underway, and the future holds great promise in the untapped potential of these groundbreaking technologies.

To address the challenge of effectively matching target models with observed images, researchers have developed a 3D vision system employing a novel technique. This innovative approach merges standard model structure descriptions with relationship-based descriptions between models using object-oriented classes [77]. The system utilizes surface models to depict geometry, with each model corresponding to an object-oriented class.

At the core of the 3D vision system lies the matching process, crucial to the understanding phase. The system employs an abstract model for relationship descriptions, encompassing various relationships such as "is-a," "has-a," aggregation, and composition aggregation. By integrating these relationships, the system gains a deeper comprehension of the objects and their interconnections within the observed scene.

The utilization of object-oriented classes enables the vision system to organize information more efficiently, mimicking how human cognition categorizes objects based on their attributes and relationships. This capability enables the system to establish meaningful links between different models, enhancing its accuracy in recognizing complex scenes [58].

By using surface models to represent geometry, this system captures detailed information about object shapes and structures, providing a high level of granularity crucial for precise matching and robust recognition, particularly in scenarios with variations or occlusions [2]. A significant advantage of this 3D vision system lies in its adaptability across various domains. Its object-oriented approach facilitates easy extension and modification to accommodate diverse objects and relationships [9], making it applicable not only in medical imaging but also in robotics, augmented reality, and computer-aided design.

The abstract model for relationship descriptions offers a higher level of abstraction, enhancing the system's resilience to noise and uncertainties commonly encountered in real-world environments. This robustness is essential for ensuring consistent and reliable performance, especially in dynamic and complex scenes [13]. Furthermore, the system's capability to handle relationships like aggregation and composition aggregation enriches its understanding of object structures. This feature is particularly valuable when dealing with interconnected components, such as mechanical assemblies or biological systems [75].

As research in computer vision and machine learning progresses, this integrated 3D vision system holds the promise of significantly enhancing the accuracy and efficiency of various applications. By combining object-oriented representation, relationship modeling, and surface-based geometry, the system has the potential to transform according how perceive and interact with the world around [24]. The development of a 3D vision system incorporating object-oriented classes and relationship-based descriptions opens up new possibilities for image understanding and recognition. Leveraging surface models and abstract relationship representations, the system achieves a more comprehensive understanding of complex scenes and objects. Its adaptability, robustness, and ability

to handle various relationships make it a versatile and powerful tool in fields such as medicine, robotics, and augmented reality [21]. As this technology evolves, expect even more exciting advancements shaping the future of computer vision and its impact on diverse industries.

The result is a concrete model with stricter limits. The following features are included in the vision system:

1. It has a model system with separate 3D and 2D models.
2. Each model uses inheritance to convey a single form notion via the is-a relationship between models.
3. An object-oriented framework is used to explain the model representation, reasoning mechanism, and picture processing.
4. A feature for comprehending incomplete line drawings is included in the system.

Furthermore, the following points are regarded as critical as model representations for comprehending pictures in such systems:

1. To reduce unique item disparities, just the essence of an object, reflecting a single notion, is employed.
2. Although shape representation is used to describe concepts, matching with the real image should not be disregarded.
3. The PART OF relation is used to express the structure clearly.
4. The approach is intended to gradually increase comprehension by explaining relationships between ideas using is-a relationships.

Knowledge representation serves as a cornerstone in various fields, aiding in the comprehension and organization of complex information. One prevalent method of knowledge representation involves the use of frames, where each frame delineates a specific topic or concept, thereby enhancing comprehensibility [57]. However, frames often function as hierarchical knowledge bases with a passive nature, which may not be conducive to the active reasoning necessary for image comprehension during the matching process.

In response to the constraints of frame-based representations, an object-oriented paradigm with active processes has emerged as a more suitable approach for image understanding. In the object-oriented paradigm, objects are construed as collections of data structures and their distinct methods, emphasizing interactions between objects and messages to develop comprehensive software solutions [15]. This paradigm facilitates concurrent data and process processing, enabling more dynamic and interactive reasoning during the matching process.

The object-oriented paradigm delineates a clear division between classes (abstract objects) and instances (concrete entities) [9]. which facilitates nuanced comparison in

AI recognition. For instance, a class may be allocated an abstract model representation encapsulating general attributes and relationships among objects within a category. Conversely, concrete objects, formed from visual cues extracted from images, correspond to specific instances of the model, embodying real-world entities.

Embracing the object-oriented paradigm empowers the vision system to proficiently utilize model representations, inference techniques, and concrete objects derived from visual data. The adaptability of object-oriented representations equips the system to navigate diverse objects and intricate scenes with heightened accuracy and efficiency [22]. The emphasis on object interactions and messages in this paradigm enhances the system's ability to actively reason during the matching process, fostering more informed decision-making based on relationships and context among objects within the scene [65].

Furthermore, the inherent capability for concurrent processing in the object-oriented paradigm enables the system to manage multiple tasks simultaneously. This concurrent processing translates into real-time performance and responsiveness, particularly beneficial in dynamic environments such as robotics, autonomous vehicles, and augmented reality, where swift and accurate decision-making is crucial [23].

The integration of an object-oriented paradigm into the 3D vision system bridges the gap between knowledge representation and image comprehension [50]. By amalgamating the strengths of both approaches, the system can effectively represent complex models, actively reason during the matching process, and translate visual cues into concrete object instances. This integration holds promise for advancing the field of computer vision and expanding its applications in diverse domains [70].

Adopting an object-oriented paradigm for knowledge representation in 3D vision systems offers numerous benefits, including active reasoning, concurrent processing, and effective handling of model representations and concrete objects [44]. This approach enables the system to grasp complex scenes and objects with greater accuracy and responsiveness. As research in computer vision progresses, the object-oriented paradigm is poised to play a pivotal role in shaping the future of image understanding and intelligent visual processing, particularly in the realm of AI recognition.

2.9. Related Work

The "Related Work" section deepens the exploration into learning pipelines and precedent studies, particularly focusing on facial recognition technology's biases. It contextualizes the significance of Joy Buolamwini's pivotal research at the Massachusetts Institute of Technology (MIT) Media Lab, which exposed substantial disparities in algorithmic performance. This seminal work, along with collaborative efforts with Timnit Gebru, has been instrumental in revealing and addressing the gender and skin-tone biases in facial recognition algorithms from leading tech companies. The section aims to provide a comprehensive background, setting the stage for the importance of the ongoing research in creating equitable and unbiased AI systems.

2.9.1. Facial Recognition Biases

In the pursuit of uncovering the accuracy and potential biases in facial recognition technologies, Joy Buolamwini, a researcher at MIT Media Lab, conducted a groundbreaking experiment [20]. She found troubling inequalities in the performance of facial recognition algorithms from prominent firms like Microsoft, IBM, and Megvii (Face++). These algorithms proved to be less effective in correctly identifying her face and gender, raising concerns about the fairness and inclusivity of such technologies [2].

To delve deeper into the issue, Buolamwini collaborated with Timnit Gebru to conduct a more extensive study [20]. They collected images of legislative branch members from six nations, with a specific focus on countries with a high female presence in politics. The dataset they compiled, known as the Pilot Parliaments Benchmark (PPB), consisted of parliamentarians from three African and three European countries [57]. The goal was to evaluate how different facial recognition APIs functioned across various skin tones.

The results of the study were startling. While the overall accuracy of the facial recognition APIs ranged from 85% to 95%, discrepancies emerged when analyzing the data across different categories. One significant finding was the difference in detecting accuracies between men and women. The algorithms performed noticeably better in recognizing male faces than female faces, indicating potential gender bias in the technology [50]. However, the most concerning disparities arose when considering skin tone as a factor. When skin tone was taken into account, the gaps in detection accuracy became much more pronounced. In this analysis, the researchers discovered that darker-skinned individuals, especially females, were the most poorly detected group by the facial recognition algorithms. On the other hand, lighter-skinned men were the best detected group.

These findings shed light on the inherent biases that exist in facial recognition technologies. The performance disparities based on both gender and skin tone highlight the need for comprehensive and unbiased datasets during the development and testing phases of these systems. Without diverse and inclusive data, facial recognition algorithms may perpetuate societal biases and reinforce unfair stereotypes [77].

The implications of these results are significant, especially in fields like law enforcement and surveillance, where facial recognition technologies are increasingly used. Misidentifications based on gender and skin tone could lead to serious consequences for individuals from marginalized communities, further exacerbating existing social injustices.

To address these issues, researchers and developers must strive for greater diversity and representativeness in the sets used to train and evaluate facial recognition algorithms. This approach can help ensure that the technologies are fair, accurate, and impartial across different demographics [67]. Additionally, transparency and accountability in the design and deployment of these systems are essential to building public trust and avoiding potential harm.

Buolamwini's research has shed light on the inequalities and biases present in facial recognition technologies. The discrepancies in detection accuracy based on gender and skin tone underscore the urgent need for more inclusive and diverse datasets and responsible development practices [20]. By addressing these challenges, allows work towards creating facial recognition technologies that are fair, and respectful of human rights for

all individuals, regardless of their gender, race, or ethnicity.

The disparities in facial recognition technology are truly alarming, as evidenced by IBM's facial recognition API detecting African women's faces with only 65.3% accuracy, while achieving an astonishing 99.7% accuracy in recognizing European men's faces. This staggering difference of 34.4% exemplifies the depth of bias present in these algorithms [58]. Such significant discrepancies in accuracy can have serious consequences, especially considering that many of these APIs are used in law enforcement.

When facial recognition technologies are employed in law enforcement, the ramifications of biases infiltrating the systems can be a matter of life or death. Misidentifications or inaccurate recognition of individuals based on gender and skin tone can lead to wrongful arrests, unjust scrutiny, and the perpetuation of harmful stereotypes [77]. Innocent people from marginalized communities may find themselves unjustly targeted due to the flawed performance of the technology. To avoid these potential harms, it is crucial for technology developers, law enforcement agencies, and policymakers to address the biases and disparities in facial recognition systems. This involves striving for more diverse and inclusive datasets, comprehensive testing across various demographics, and transparent reporting of algorithm performance [70]. By holding the developers accountable and promoting responsible use of the technology, helps towards creating a more equitable and just society where the use of facial recognition does not perpetuate existing social injustices.

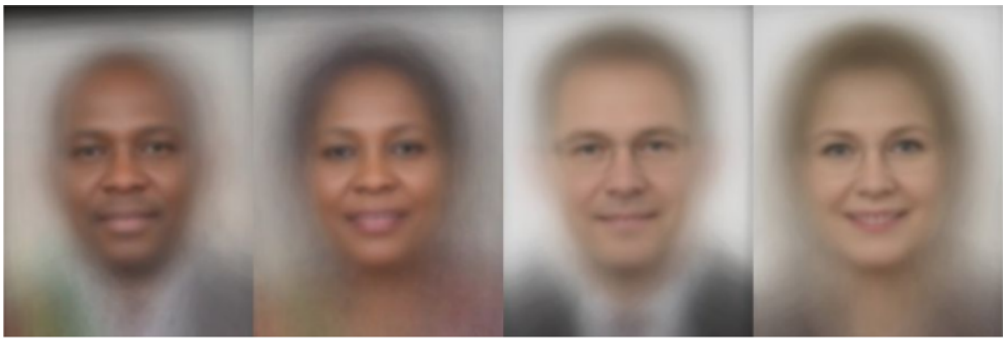

Figure 2.13.: An examination of biases in facial recognition: This image represents an average composite of faces generated by IBM's facial recognition to demonstrate the variance in accuracy and representation across different demographics.[55]

The study's findings highlight many crucial insights:

1. The quality and diversity of the training data substantially influence an algorithm's performance. This underlines the importance of a broad training dataset in order to prevent biases and obtain higher accuracy.

2. Simply looking at aggregate statistics may not reflect the real depth of the datasets biases. To successfully identify and correct biases, data must be analyzed across sev-

eral groupings. Furthermore, this prejudice is not restricted to a single corporation; it pervades the whole industry.

3. The figures in the study are not set and may vary over time. Companies are actively working on reducing biases from their sets, as seen by the large differences between 2017 (Figure 2.14) and a following 2018 research (Figure 2.15). This demonstrates a commitment to progress.

4. Independent research and public benchmarking can encourage industry-wide changes. The researchers advise commercial firms to eliminate biases and aim for better results by putting them to testing using public benchmarks.

	Overall	Darker Female	Darker Male	Lighter Female	Lighter Male
Microsoft	93.70%	79.20%	94.00%	98.30%	100%
Face++	90.00%	65.50%	99.30%	94%	99.20%
IBM	87.90%	65.30%	88.00%	92.90%	99.70%

Figure 2.14.: Highlighting facial recognition disparities, this 2017 data chart reveals lower accuracy rates for darker-skinned females and higher rates for lighter-skinned males, showcasing the need for more balanced AI systems.[55]

	Overall	Darker Female	Darker Male	Lighter Female	Lighter Male
Microsoft	99.52%	98.48%	99.67%	99.66%	100%
Face++	98.40%	95.90%	98.70%	99%	99.50%
IBM	95.59%	83.03%	99.37%	97.63%	99.74%

Figure 2.15.: The 2018 data chart illustrates significant improvements in facial recognition technology, with a notable increase in accuracy across all demographics, especially for darker-skinned females, reflecting advancements in AI fairness.[55]

2.9.2. Object Detection Biases

The study's main findings highlight significant disparities in object classification APIs' performance based on the geographic location of the photographs. Figure 2.16 illustrates that the APIs performed notably worse in images from lower-income regions. The lack of representation from Africa, India, China, and Southeast Asia in databases like

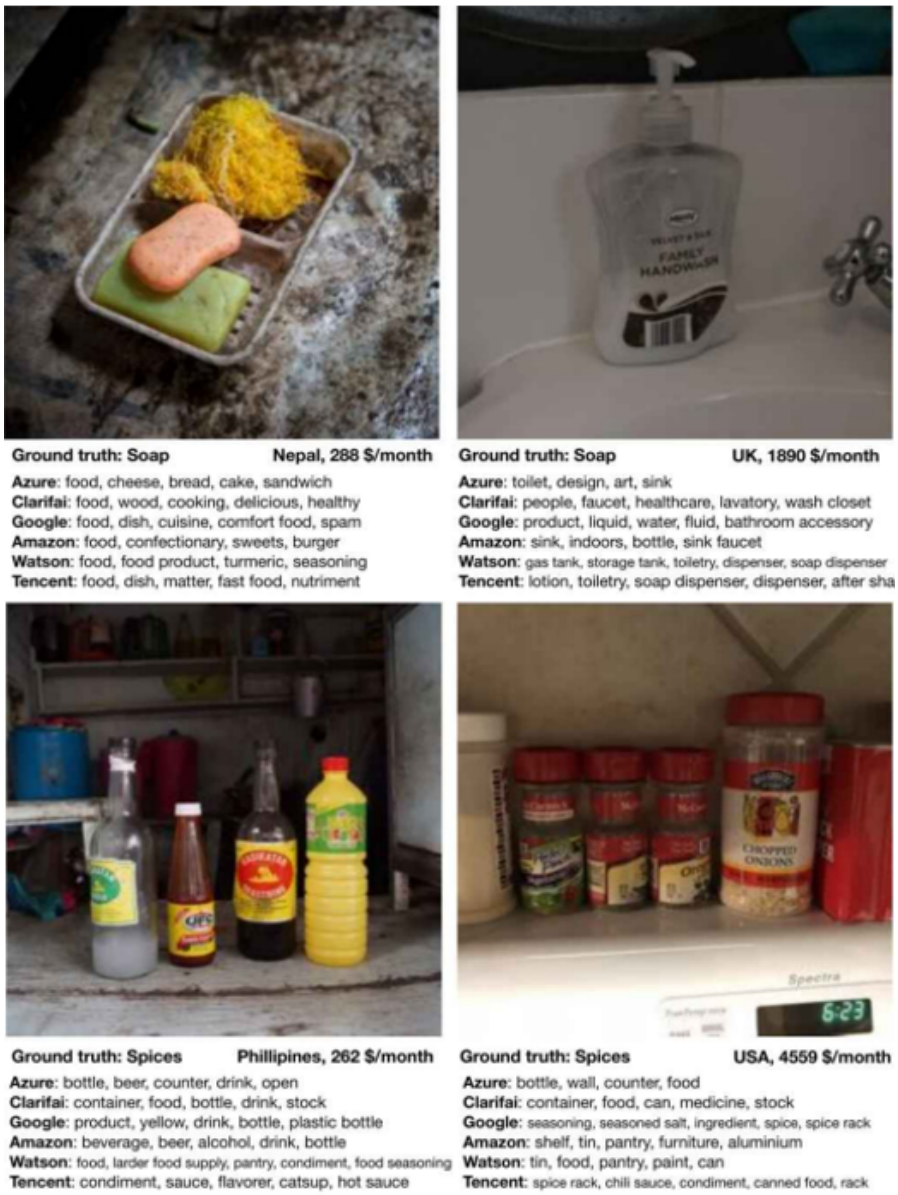

Figure 2.16.: The image reveals a significant classification bias by various cloud services, e.g a soap is mistakenly identified as food-related items such as cheese, bread, and noodles. This example underlines the potential inaccuracies in AI object recognition systems.[55]

ImageNet, COCO, and OpenImages contributes to the poor performance of these APIs on non-Western images.

One of the crucial factors contributing to this discrepancy is the limited coverage of photographs from these regions in the sets used to train the APIs [9]. The sets primarily

consist of images collected through keyword searches in English, which results in the exclusion of photos described using phrases in other languages. This language bias further contributes to the under representation of non-Western images, leading to inaccurate and biased results as in Figure 2.17.

To address these issues, there is a pressing need for more diverse and inclusive datasets that encompass a wide range of geographic locations and languages. Including a broader variety of images from different parts of the world will lead to improved performance and reduced biases in object classification APIs [61]. Furthermore, researchers and developers should prioritize collecting and curating datasets that better reflect the global diversity of images, ensuring that these technologies are fair, accurate, and representative for users from all backgrounds and regions.

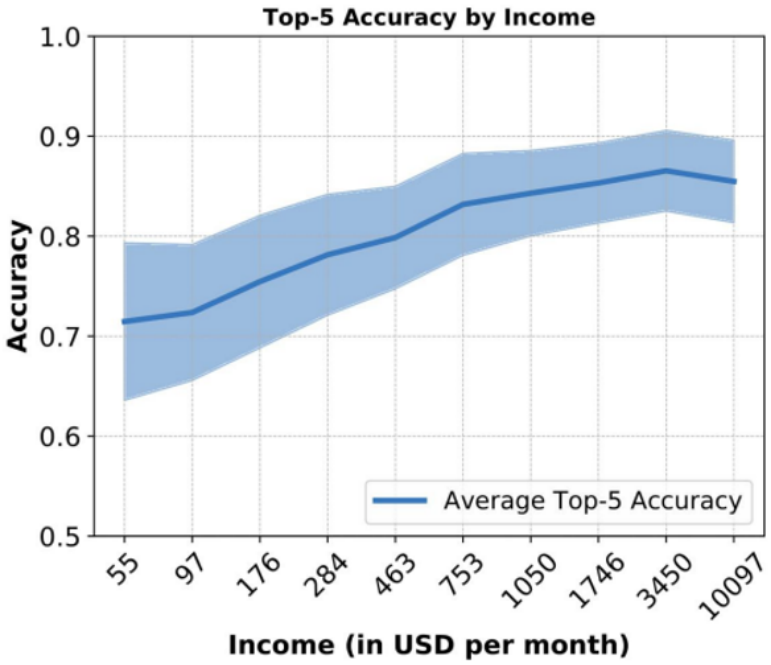

Figure 2.17.: This graph illustrates the correlation between income levels and the top-5 accuracy of an AI system, revealing a trend where higher income brackets correspond with greater accuracy.[55]

Burrell [21] discovered three levels of opacity in the image taggers studied in his study. For starters, the algorithms utilized in photo taggers are proprietary, and their creators reveal nothing about their decision-making processes or how they produce picture tags. Because these algorithms employ deep learning, it is technically difficult to provide meaningful explanations, and even if explanations are provided, many consumers may lack the technical skills to grasp them. As a result, image taggers have become "power brokers" [24], as they carry out different everyday operations autonomously without human involvement or oversight [22]. Furthermore, these algorithms are frequently seen

as objective by users, and some may be ignorant of their presence or use in systems [31].

The study's findings [58, 54, 9, 24] show that image tagging algorithms are not unbiased, particularly when processing photographs of humans. While there was no indication of tags reflecting negative opinions on physical appearance, several algorithms created positive tags such as "attractive," "sexy," and "fine-looking." Both developers and end users may be surprised by this result.

The opacity and lack of transparency in the functioning of picture taggers raise concerns about the potential biases embedded within these algorithms. The algorithms' proprietary nature and limited disclosure of decision-making processes make it difficult for researchers and users to scrutinize and understand the mechanisms behind the generated tags [70]. This lack of transparency can lead to unintended and potentially harmful consequences, such as reinforcing societal beauty standards or perpetuating stereotypes based on attractiveness.

Furthermore, the autonomous nature of picture taggers, operating without human oversight, makes it challenging to identify and correct biased outcomes. The algorithms' perceived objectivity by users may lead them to trust and rely on the generated tags without questioning their accuracy or fairness [10]. To address these issues, there is a need for greater transparency and accountability in the development and deployment of picture tagging algorithms. Developers should work towards providing more accessible and understandable explanations for the generated tags, especially in cases involving images of humans [61]. Efforts to mitigate biases and ensure fairness should be prioritized, and the datasets used to train these algorithms should be carefully curated to avoid perpetuating harmful stereotypes or unfair judgments.

The opacity and lack of transparency in picture tagging algorithms raise significant concerns about potential biases and unintended consequences. The discovery of positive tags related to attractiveness in human images emphasizes the need for greater scrutiny and responsible development of these algorithms [15]. By promoting transparency, fairness, and inclusivity, ensures that picture taggers are unbiased and respectful of human diversity and individuality.

The correlation of certain physical attributes with beauty tags in the algorithms raised concerns as it perpetuated the notion that physical attractiveness is associated with positive traits ("beautiful is good"). Even after accounting for the ethnicity and gender of the individuals in the photographs, the algorithms still tended to associate tags expressing positive emotional sentiment with images of more attractive individuals, as judged by humans [22]. This brought about new questions about the impact of physical attractiveness on the application of positive emotion tags.

2.9.3. Cost Implications of Using APIs

In terms of cost [77] conducted a comparison of the prices of five APIs, assuming heavy-duty usage at approximately 1 query per second (QPS) for an entire month, totaling around 2.6 million requests per month. The cost analysis is essential as it provides insights into the economic implications of using these APIs in various applications and services.

Understanding the cost factors helps developers and businesses make informed decisions when integrating these APIs into their systems.

By considering both the biases in the algorithms' outputs and the economic implications of using picture tagging APIs, researchers and developers can work towards creating more equitable and responsible AI systems. Addressing biases and promoting fairness in the development and deployment of these algorithms is vital to ensure that they serve all users fairly and accurately, without perpetuating harmful stereotypes or reinforcing societal biases. Additionally, cost analyses enable better resource planning and optimization, allowing organizations to make cost-effective choices while maintaining the ethical use of AI technologies.

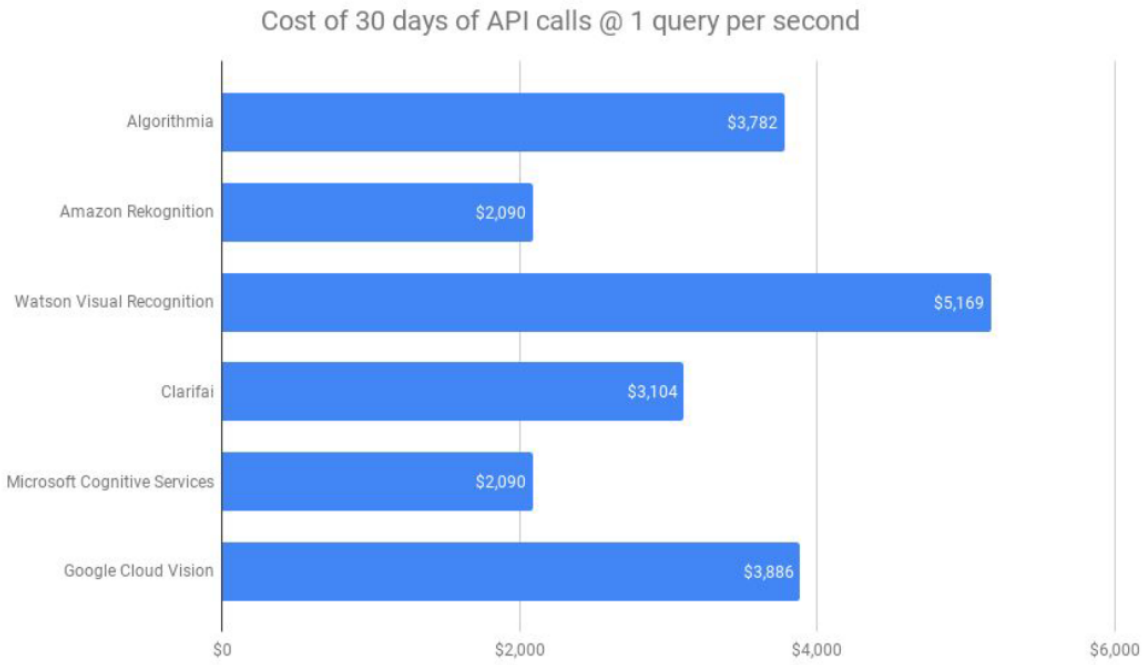

Figure 2.18.: Comparative Analysis of API Costs: This bar graph illustrates a cost analysis of various cloud-based vision APIs over a 30-day period, assuming a consistent call rate of 1 query per second. It allows for a clear comparison of the expense associated with using different API providers.[55]

When considering the cost of using picture tagging APIs, it is important to acknowledge that extensive consumption may primarily apply to large organizations. However, for many developers, the fees for API usage may be considered insignificant. Most cloud providers offer a free tier with a predetermined number of calls per month, typically 5,000 for all providers except Google Vision, which offers 1,000 free calls per month. Beyond the free tier, the cost is approximately $1 per 1,000 calls as shown in Figure 2.18.

2.9.4. Accuracy in Text Extraction Using COCO-Text

To assess the accuracy of various service providers, consistent measures for comparing their performance on external datasets are crucial. The [77] conducted an analysis of text extraction quality using the COCO-Text dataset, which is a subset of the MS COCO dataset. This set comprises 63,686 photos containing text in real-world environments, such as banners, street signs, buses, store price tags, and garment labels [65]. Due to the challenging nature of this set, it serves as a robust benchmark for evaluating the performance of different APIs.

In this analysis, the Word Error Rate (WER) is used as the benchmarking statistic, which focuses on whether a word is accurately retrieved regardless of its location (i.e., bag of words) [23]. A match is considered when the complete word is accurately retrieved. For the COCO-Text validation dataset, images containing one or more instances of readable text (full-text sequences without pauses) are selected, and text examples of more than one character length are compared [83]. These photos are then submitted to several cloud vision APIs, with the results depicted in Figure 2.19.

The use of a standardized benchmark like COCO-Text enables researchers and developers to make objective comparisons of different cloud vision APIs. It provides valuable insights into the algorithms' performance when faced with real-world text in diverse and challenging settings [13]. By using consistent measures and benchmark datasets, the evaluations become replicable and reliable, allowing for more informed decisions in selecting the most suitable API for specific applications.

The results of the analysis shown in Figure 2.19 offer valuable information about the strengths and limitations of various cloud vision APIs in text extraction tasks. These findings can guide developers and organizations in choosing the most appropriate API based on their specific requirements, accuracy needs, and cost considerations [54].

The assessment of cloud vision APIs' accuracy is essential to ensure their effective and responsible use. Standardized benchmarks, such as COCO-Text, play a critical role in evaluating the algorithms' performance on external datasets. By employing consistent measures and comprehensive evaluations, help to make more informed decisions about API selection, leading to improved efficiency and better outcomes in various applications involving text extraction from images.

The results obtained from the challenging COCO-Text set are truly remarkable, as most state-of-the-art text extraction algorithms from previous years could not achieve accuracy levels above 10%. This underscores the significance and effectiveness of deep learning algorithms in handling complex text extraction tasks. The continuous improvement in the performance of various cloud-based APIs, as revealed by Matsangidou and Otterbacher [61], further emphasizes the ongoing advancement of these systems.

Matsangidou and Otterbacher [61] used voting modules in their experiment to examine the accuracy of five cognitive API-based models and three voting modules. The experiment took done in a community laboratory setting, where images were taken on a regular basis with a fixed USB camera and sent to a server. These 1280x1024 pictures were taken between July 2018 and March 2019 [24]. The researchers chose seven fine-grained settings for the experiment and labeled 100 example photographs for each scenario, yielding a

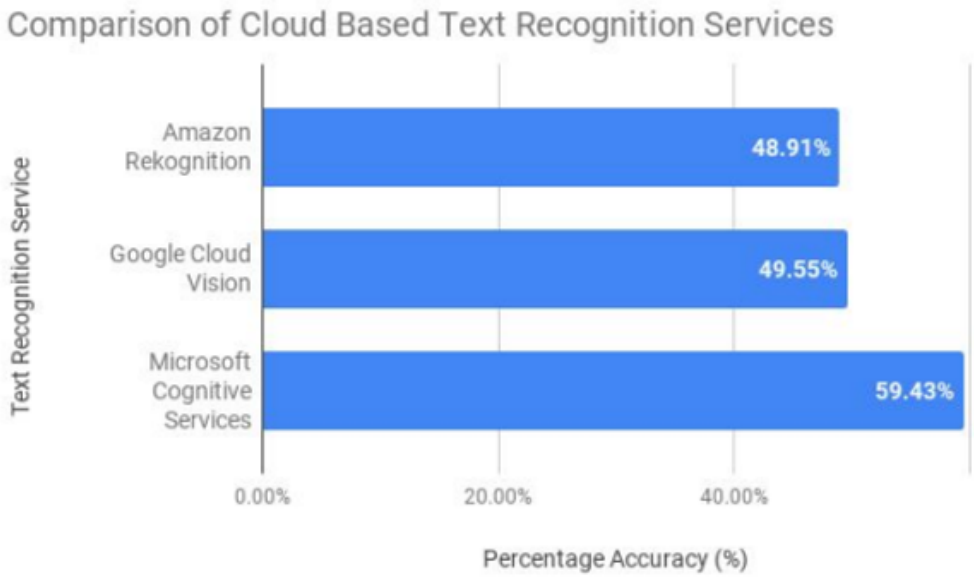

Figure 2.19.: Assessing API Performance: The bar chart presents a Word Error Rate comparison among text extraction services as of August 2019, showing the percentage accuracy for different APIs.[55]

total of 700 images. These photos were then separated into training and test sets at random.

The use of voting modules in the experiment enables a more robust evaluation of the cognitive API-based models. By combining the results of multiple APIs through voting, the researchers can obtain a more reliable and accurate assessment of text extraction performance. This approach helps mitigate potential biases or limitations in individual APIs and provides a comprehensive view of their capabilities.

The communal laboratory environment and the collection of photos over several months ensure a diverse and representative dataset for the experiment. This real-world data capture allows for the evaluation of the algorithms' performance in various practical scenarios, providing insights into their adaptability and effectiveness in handling different contexts [17]. The fine-grained settings and careful tagging of representative photos contribute to the richness and quality of the set. By focusing on specific contexts, the experiment can offer detailed insights into how well the APIs perform in different scenarios and under varying conditions [67].

Overall, the experimental design and dataset selection play a crucial role in the validity and reliability of the study's findings. The use of challenging sets, voting modules, and real-world scenarios ensures a comprehensive evaluation of the cognitive API-based models' accuracy and performance [27]. The continuous improvement observed in cloud-based APIs reflects the ever-evolving nature of these systems, indicating promising advancements in the field of computer vision and image understanding.

The experiment conducted in [61] provides valuable insights into the accuracy and performance of cognitive API-based models for text extraction tasks. The use of challenging sets and voting modules, along with careful dataset selection, strengthens the validity and significance of the study's findings. As cloud-based APIs continue to improve, their potential for enhancing text extraction from images in real-world scenarios becomes increasingly promising.

In the experiment conducted in [61], various APIs were utilized to create the recognition model for seven different scenarios. The APIs used included the Microsoft Azure Computer Vision API, IBM Watson Visual Recognition API, Clarifai API, Imagga REST API, and ParallelDots API. Each API was tasked with detecting similar photographs from various viewpoints, and the resulting tags were vectorized using Term Frequency-Inverse Document Frequency (TF-IDF) [23]. The data and labels were then imported into Microsoft Azure Machine Learning Studio, and classifiers were created for each cognitive API using the Multiclass Neural Network.

Among the five models, the one based on the Imagga API exhibited the highest accuracy, achieving an impressive score of 0.9429. On the other hand, the model based on the ParallelDots API had the lowest accuracy, with a score of 0.7718. The performance of the five models varied across different contexts, with each model excelling in recognizing specific circumstances [9].

To further enhance the accuracy of the recognition model, the researchers incorporated three voting modules. These voting modules resulted in a substantial increase in overall accuracy. The majority voting strategy achieved an accuracy of 0.9753, the score voting strategy had an accuracy of 0.9776, and the range voting strategy performed even better, obtaining the highest accuracy of 0.9833 within the range of 0.5 to 0.6 [13]. When examining the accuracy of each voting strategy in different contexts, the range voting approach demonstrated the most consistent and overall higher accuracy. However, in certain circumstances, such as "Play games" and "General meeting," the range voting strategy exhibited slightly lower performance compared to other contexts.

The utilization of voting modules showcases the effectiveness of combining multiple cognitive APIs to obtain more accurate and robust results. The majority voting strategy leverages the collective decisions of all APIs to arrive at a final decision, while the score voting strategy considers the confidence levels assigned by each API to the predictions [76]. The range voting strategy, which considers a specific range of confidence scores, demonstrated exceptional consistency and accuracy across various contexts.

The experimental findings highlight the importance of selecting the appropriate cognitive API based on the specific context and use case. Different APIs may excel in recognizing particular scenarios, and their performance can be further improved by employing voting strategies.

In [61] experiment provides valuable insights into the performance of various cognitive APIs and the effectiveness of voting modules in enhancing accuracy. By combining multiple APIs and adopting appropriate voting strategies, developers and researchers can create recognition models that deliver more reliable and precise results across diverse contexts and applications [23]. As the field of computer vision continues to advance, these

approaches hold great promise for improving image understanding and furthering the capabilities of AI systems in real-world scenarios.

2.10. Summary

Chapter two provides overview of several Application Programming Interfaces APIs, focusing specifically on Clarifai, Microsoft Cognitive Services, and Amazon Rekognition. The primary objective of this study is to examine the capabilities and implications of these application programming interfaces APIs, shedding light on their performance, functionality, and potential consequences. Furthermore, the chapter highlights the need of doing an analysis of software quality when considering the utilization of these application programming interfaces APIs. The need of incorporating data-driven insights is emphasized in order to make informed assessments about their suitability for certain applications.

The study also examines recent developments in face recognition research, specifically emphasizing the operational principles of Convolutional Neural Networks CNNs. This study examines several deep learning architectures commonly employed in the industry and imparts fundamental insights into the technological underpinnings of these application programming interfaces APIs. Additionally, the chapter thoroughly explores the sets employed in deep learning applications, delving into the crucial subject of bias and its consequences. Overcoming bias in AI systems is of utmost importance in order to maintain justice and uphold ethical standards.

This chapter offers a comprehensive examination of deep learning technology, encompassing an analysis of the capabilities of popular application programming interfaces APIs and an exploration of the fundamental principles behind Convolutional Neural Networks CNNs. This comprehension provides a solid foundation for the subsequent chapters, that delves into specific case studies and applications.

3. Methodology

In this section, we elaborate on the processes involved in our methodology, starting with the data collection and preprocessing stages. We will then discuss how we identify and select target websites for web scraping, as well as the criteria for choosing an appropriate web scraping library. Following this, we will delve into the vectorization and tokenization aspects, utilizing the FastText model [49]. A detailed explanation of the FastText model's mechanisms will also be provided. Lastly, we will describe the architecture of the decision support tool and explain how the pipeline operates using these libraries.

3.1. Data Collection and Processing

Efficient data collection and processing strategies are crucial when conducting research on internet resources such as official websites and reference pages for Image Recognition APIs [33]. This section provides a comprehensive analysis of these essential processes. The procedure encompasses the systematic identification of target websites, the deliberate utilization of web crawling technologies, and the subsequent phase of data analysis. Collectively, these methodologies ensure the acquisition of reliable and relevant data, serving as the fundamental basis of robust research in the dynamic realm of digital information.

3.1.1. Data Collection

The preliminary stage of data collection in research relies on online resources as a crucial foundation. This section explores the approaches and tactics utilized during this period, establishing the foundation for the subsequent analytical processes.

Locating the Target Websites

The process begins by carefully identifying and examining official webpages or documentation pages related to the Image Recognition APIs being investigated. An exemplary illustration of this procedure can be observed at the official website of the Google Cloud Vision API, accessible via the following link: cloud.google.com/vision/. This initial undertaking is akin to marking coordinates on a map, forming the essential basis for the complete framework of data collection.

The significance of discovering and accessing these digital stores of information cannot be overstated. Online platforms serve as repositories of extensive information, including comprehensive documentation for API functionalities, pricing models, limitations, and

user insights [80, 85]. Accurately finding and accessing these sources is like discovering valuable repositories, vital for study in the era of digital information.

Selecting a Web Scraping Implementation Library

Python, a highly adaptable and extensively utilized programming language, provides a large range of web scraping tools. Among these tools, two notable options are BeautifulSoup and Scrapy, each possessing specific advantages [78]. The selection of the library depends on the particular goals and extent of the data collection activity.

- **BeautifulSoup**, a lightweight and user-friendly library, is considered an ideal option, especially for individuals new to web scraping. The library offers an optimized interface for parsing HTML content and extracting data in a straightforward manner [1]. This tool is suitable for relatively small-scale scraping activities, making it a good choice for beginners.
- **Scrapy**, a robust and comprehensive framework, is designed for large and complex web scraping endeavors. It not only facilitates data extraction but also enhances web crawling capabilities, making it the preferred tool for extensive and complex scraping projects [72]. When dealing with massive API documentation and the need for thorough data collection, Scrapy proves to be a reliable companion.

Accessing and Extracting Information

Once the suitable library has been established, the next step is utilizing its functionalities to interact with the relevant API documentation pages. This involves the strategic coordination of HTTP requests directed towards the designated websites to get essential data components [35]. These components cover extensive information, including detailed insights into API functionality, complex pricing structures, usage restrictions, and important user feedback. The crucial data bits are carefully extracted from the HTML content through thorough evaluation.

To exemplify this procedure, consider the use of BeautifulSoup to analyze the Google Cloud Vision API (see Python Code 3.1). This application can streamline the process of extracting the table of contents from the documentation page, serving as a vital navigational aid. The table of contents facilitates efficient navigation to particular areas of the documents that are significant for the research.

```
# Make a request to the website
response = requests.get('http://cloud.google.com/vision/')

# Parse the HTML content
soup = BeautifulSoup(response.text, 'html.parser')

# Extract the title of the webpage
title = soup.title.text

print("Webpage Title:", title)
```

Listing 3.1: Python BeautifulSoup example

The data collection phase is the initial stage of the expedition, marking the commencement of a quest to extract information from the digital domain [78]. This comprehensive study is characterized by meticulousness and deliberation, serving as the foundation for a deeper exploration of the complex realm of Image Recognition APIs and their various aspects.

3.1.2. Data Processing

Data cleansing, also referred to as data scrubbing or data cleaning, constitutes a crucial stage within the data processing workflow. The primary objective is to enhance and standardize the collected data, ensuring its uniformity, consistency, and suitability for future research endeavors [82, 81]. This meticulous method encompasses several primary objectives:

1. **Elimination of Superfluous Data**: The primary objective of data cleansing is to eliminate unnecessary or non-essential elements from the dataset. This process results in a more concentrated and significant set by removing extraneous noise.

2. **Text Layout**: Measures are taken to regulate the layout of textual material, ensuring adherence to established standards. This consistency enhances the uniformity of text presentation, minimizing disparities that might impede further analysis.

3. **Data Structuring**: The data is systematically and logically structured to facilitate further analysis. This may involve creating organized tables or datasets to enhance data accessibility and utility.

During data purification, diligent efforts are made to remove any remnants of HTML components identified within the gathered text. This approach ensures the removal of web-based formatting remnants that might compromise the data's integrity [84]. Additionally, the text is systematically converted to lowercase, a straightforward yet effective procedure that reduces mistakes and ensures content consistency.

3.2. FastText Model

The FastText model, introduced by Facebook AI Research (FAIR), is a highly efficient and effective method for text classification and word representation learning. It extends the traditional bag-of-words model by incorporating subword information, particularly using character-level n-grams. This allows FastText to capture the morphology of words, providing better representations for uncommon words. It is noted for its speed and accuracy in classifying large datasets and handling multiple languages, making it a versatile tool in natural language processing tasks. The model achieves a good balance between resource efficiency and performance, suitable for various NLP applications [49].

3.2.1. FastText Model Architecture

The FastText model employs a simple yet effective architecture for text classification [49]. It represents sentences using a Bag of Words (BoW) approach and extends it by incorporating a bag of n-grams as additional features. This aids in efficiently capturing some word order information. The model architecture includes a weight matrix that acts as a look-up table for words, followed by averaging the word representations into a text representation. This text representation is then fed to a linear classifier. FastText uses a hierarchical softmax based on a Huffman coding tree for computational efficiency, especially when dealing with a large number of classes. This hierarchical structure significantly reduces computational complexity during training and testing.

3.2.2. Vectorization using FastText Model

Transforming unstructured text into meaningful, analyzable data is a challenge in machine learning, especially within the realm of natural language processing (NLP). A fundamental aspect of this transformation is feature vectorization, which converts textual information into a format suitable for machine learning algorithms. The FastText model, developed by Facebook AI Research, emerges as a significant tool for feature vectorization. Unlike conventional methods, FastText goes beyond mere word-level analysis and delves into the subword level, considering character n-grams within words. This enables the model to capture the morphology of words, enriching the representation of text data, particularly for languages with rich and complex word formations. The FastText approach excels in understanding and representing the nuances of word usage in different contexts, enhancing the accuracy of text analysis. This thesis leverages the sophisticated capabilities of the FastText model for tokenization and feature vector generation. The following section provides a comprehensive exposition of the fundamental principles and mathematical underpinnings of the FastText model, elucidating how it revolutionizes the process of converting raw text into actionable data for machine learning applications [49].

3.3. Tokenization

Tokenization is a fundamental component wherein text data is segmented into discrete units, such as individual words or phrases. Tokenization enables the manipulation of discrete units within a text, facilitating subsequent computational operations [68]. Several strategies for tokenization exist; however, one widely used and pragmatic technique is the use of regular expressions to divide text by whitespace, efficiently separating it into individual elements and removing duplication.

For example, consider the statement: "Text data is analyzed by machine learning algorithms." Tokenization of this statement results in the identification of the following terms: "Text," "data," "is," "analyzed," "by," "machine," "learning," "algorithms."

3.4. Scores Calculation

After tokenizing the text input, the next critical step is calculating the similarity scores for each word. This score is a metric that evaluates the significance of a phrase in the text by considering the Euclidean distance between the words as vectors and a set of target vector representations from a word dictionary for each category (e.g., performance, cost, limitation, and functionality). For instance, the cost category includes words like 'currency,' 'per month,' 'rates,' and 'cost.' This evaluation is performed using cosine similarity.

Cosine similarity measures the cosine of the angle between two non-zero vectors (representing words or sentences) in a multi-dimensional space. These vectors typically represent words or sentences, allowing us to determine their similarity. This method is widely used in applications such as document retrieval.

The similarity score, $similarity(A, B)$, between two words A and B can be calculated as:

$$similarity(A, B) = \frac{\sum_{i=1}^{n}(A_i \times B_i)}{\sqrt{\sum_{i=1}^{n}(A_i)^2} \times \sqrt{\sum_{i=1}^{n}(B_i)^2}} \tag{3.1}$$

where A_i and B_i are the components of the vector representations of the two words that are output from the FastText model, such that both of them are equal in dimensions[32].

3.5. Clustering Analysis

Clustering analysis is a powerful unsupervised technique that aims to impose structure on unorganized data within the expansive domain of machine learning. Through the amalgamation of pertinent data fragments, this methodology facilitates the identification of patterns and associations that may elude the casual observer [60, 73]. In our study, clustering was performed as follows: after calculating the similarity scores, we determined whether these scores exceeded a certain threshold. Words that surpassed this threshold were selected for further analysis. Specifically, when employing the maximum method, the threshold range was set between 0.8 and 0.9 (out of a maximum of 1). Alternatively, when using the mean and median methods, the threshold was set at 3.4. Establishing the optimal threshold proved to be a challenging task. It required analyzing a substantial amount of data, calculating mean methods with the corresponding similarity scores, and then determining the highest achievable value. We then slightly reduced this value to identify the most suitable threshold for selection.

3.6. Summary

This section presents a concise overview of the methodology used in research. It begins with data collection and processing strategies, focusing on gathering information from internet resources, especially official websites related to Image Recognition APIs. It details the selection of target websites and the use of web scraping libraries, comparing

BeautifulSoup's suitability for smaller projects and Scrapy's capabilities for larger, complex tasks. It then delves into data processing, emphasizing the importance of data cleansing, including the removal of unnecessary data, standardization of text layout, and structuring of data for analysis. Following this, the FastText model is introduced for text classification and word representation, highlighting its architecture and the use of Bag of Words and n-grams. The process of tokenization is discussed, describing how text is segmented into individual elements using regular expressions. The section concludes with a discussion on calculating similarity scores for word significance and the methodology for clustering analysis, including the challenges of establishing an optimal threshold for data selection.

Overall, this part methodically outlines the steps from data collection to analysis, incorporating advanced tools like the FastText model in the research process.

4. Results and Discussion

This thesis dissects and compares six leading image recognition cloud services: Google Cloud Vision API, Amazon Rekognition, Microsoft Azure Computer Vision API, IBM Watson Visual Recognition, Clarifai, and Imagga. Each of these services offers a unique set of features. The heart of this analysis lies in understanding these platforms beyond their marketing gloss – delving into their performance, cost, functionality, and limitations. Such an understanding is pivotal not just for their practical deployment but also for aligning them with specific user requirements.

Our approach, powered by a comparison support tool, gathered and filtered data from each service's documentation on their websites. This exercise was not about amassing data in volume but about stratifying it into the core categories that matter: performance, cost, functionality, and limitations. The comparison support tool, utilizing vectorization techniques and similarity analysis, has been instrumental in uncovering the subtle yet critical distinctions and parallels among these services.

The subsequent sections aim to reveal the outcomes of our thorough analysis. We delve into the complexities of each service, offering an in-depth examination of their functionalities. This part of the thesis explores the assessment of each API, providing a comparison based on the previously mentioned criteria. It also highlights the distinctions between each API, detailing the data categorization and, ultimately, the results. This will include an interpretation of the final results and their significance.

4.1. APIs Evaluation

APIs for image identification come in a variety of flavors, each with its own set of features, capabilities, pricing structures, performance indicators, and limitations. Selecting the best API for a certain project can be a difficult task. The following part provides an overview of each API mentioned above.

4.1.1. Google Cloud Vision

A major component of the Google Cloud Vision API is the exploitation of Google's powerful machine learning algorithms, which enables the provision of various capabilities for picture recognition [14]. The system can recognize and classify a wide variety of visual features, including people and entities, as well as language information and written representations of those elements. The system includes optical character recognition (OCR), label identification, and explicit content detection.

The Google Cloud Vision API is known for its extraordinary capabilities, particularly in object and text recognition. When applied to a wide range of applications, the technique

demonstrates precision and versatility [14]. The pricing methodology for accessing the Google Cloud Vision API is determined by the total number of queries run in conjunction with the specific functions invoked. The free tier of the platform attempts to facilitate job execution on a smaller scale by offering a subset of the platform's functionality. Users can gain access to a wide range of services and features through Google's API, including object identification, facial recognition, and OCR. Despite its many benefits, the Google Cloud Vision API may not be the most cost-effective solution for applications requiring extensive user engagement.

4.1.2. Amazon Rekognition

Amazon Rekognition is a cloud-based technology offered by Amazon capable of analyzing photographs and videos. The system can detect and classify various components, including static or dynamic audio-visual content. This includes both moving images and still images, covering goods, people, places, languages, attitudes, and other elements [74]. It also incorporates facial recognition capabilities into its operating mechanism.

Applications requiring effective image and video processing may benefit from Amazon Rekognition's consistent performance. The pricing for Amazon Rekognition is based on the volume of data processed and the number of queries run. The software provides a free layer of service, but there are some restrictions, making it suitable for small businesses. Amazon Rekognition's functions include celebrity recognition, face identification, and emotional state detection [74]. While effective, Amazon Rekognition may not be the most cost-efficient choice for high-traffic applications.

4.1.3. Microsoft Azure Computer Vision

Microsoft's Azure Computer Vision API can perform various tasks, including image analysis, OCR, face detection, and the identification of adult content [79]. Images are highly valuable in document analysis as they may contain extractable important information.

The Microsoft Azure Computer Vision API exhibits a high level of efficacy and accuracy in image recognition tasks. It displays adaptability across a wide variety of applications [79]. The pricing model for using the Azure Computer Vision API varies based on the number of transactions processed and the specific functions called upon. Access is provided to a free tier with a limited set of features. Image description, text extraction, and handwriting recognition are some of the functions offered by the Azure Computer Vision API, demonstrating its flexibility across various application use cases. While promising, the Azure Computer Vision API may not be the most economically efficient solution for frequent searches.

4.1.4. IBM Watson Visual Recognition

IBM Watson Visual Recognition aims to facilitate the training of specialized models for improved image recognition [12]. The system can recognize and categorize various elements,

including human facial features, spatial placements, and relationships. Additionally, it can create individualized classifiers using user-provided visual input.

IBM Watson Visual Recognition offers great performance and adaptability for various applications. This method performs exceptionally well when individualized models are required [12]. The pricing model considers the number of custom classifiers and the number of images used. During the trial period, clients can try out various paid services at no additional cost. This method's ability to build unique classifiers tailored to specific image recognition tasks is its key benefit. However, generating unique models may require a larger commitment of both time and money resources compared to other APIs.

4.1.5. Clarifai

The Clarifai image recognition API offers significant adaptability. The system can recognize a wide variety of properties within a snapshot, including colors, objects, scenarios, and abstract thoughts [16]. The software is praised for its many functions and ease of use.

Clarifai's remarkable performance makes it ideal for applications requiring effective image processing. Its pricing model is based on the quantity of operations carried out and the accuracy of forecasts [16]. There is a free trial version with restricted features and functions. The Clarifai API allows for modifying the training of models and is known for its intuitive user interface and design tailored to programmers' needs. Despite its dominance in the market, Clarifai may not always be the most cost-effective choice for large-scale operations.

4.1.6. Imagga

Imagga's platform effectively classifies and categorizes images [37]. The method accurately detects and categorizes a wide variety of things, situations, and ideas depicted in photographs, making it well-suited for content categorization.

Imagga performs image tagging and recognition efficiently. Its price structure is determined by the number of photographs uploaded and the number of API calls made [37]. Users have limited access and permissions during the trial period. Due to its expertise in organizing and annotating photographic images, Imagga is well-suited for managing visual content. Despite widespread recognition, Imagga's selection of customization options is fairly restricted compared to other APIs.

Given the numerous options available, an exhaustive selection process is necessary to determine the best image recognition APIs for specific project requirements. Evaluating APIs requires an in-depth analysis of their performance, cost, practicality, and constraints. Our Python program intends to make the review procedure more efficient by providing a methodical framework for removing, enhancing, and contrasting API documentation. Image recognition systems achieve higher operational efficiency when they can make accurate decisions.

4.2. Comparison Criteria

The primary objective of this comparative analysis is to dissect and understand the diverse capabilities and offerings of prominent image recognition cloud services in the current technological landscape. The services under scrutiny in this study are Google Cloud Vision API, Amazon Rekognition, Microsoft Azure Computer Vision API, IBM Watson Visual Recognition, Clarifai, and Imagga. Each of these services is a front-runner in the field of image recognition, yet they differ in various aspects that can significantly impact their suitability for specific applications.

The comparison is structured around four key aspects:

1. **Performance:** Assessing the accuracy and efficiency of each service in processing and recognizing images. Performance is a critical factor, as it directly influences the reliability and applicability of the service.

2. **Cost:** An essential consideration for any user, cost analysis encompasses not just the direct expenses but also the cost-effectiveness of each service. This aspect evaluates the pricing structures and how they align with the features offered.

3. **Functionality:** Examining the range of features and capabilities each service offers. Functionality includes the variety of image recognition tasks each service can perform, the ease of integration into existing systems, and the flexibility of the API to serve specific needs.

4. **Limitations:** Uncovering and articulating the limitations inherent in each service, whether in terms of image processing capabilities, scalability, or specific use-case restrictions.

By analyzing the API services against these four parameters, this study seeks to provide a comprehensive overview that aids in discerning the most suitable image recognition cloud service for different user needs and scenarios. The outcome is intended to serve as a guide for both practitioners in the field and businesses looking to integrate image recognition technologies into their operations, offering insights that are both practical and theoretically grounded.

This comparative analysis is not just a juxtaposition of features and numbers; it is an exploration aimed at uncovering deeper insights into the operational dynamics of these leading cloud services, providing a clearer understanding of their place in the current and future technologies of image recognition APIs.

4.2.1. Performance

In the realm of technology, APIs have become indispensable in sectors like e-commerce, healthcare, autonomous vehicles, and security systems. These technologies largely hinge on image recognition, which includes object identification and pattern classification [71, 56]. The performance of an API in this context is paramount, as it directly impacts

its effectiveness and accuracy in processing and analyzing images. Table 4.1 summarizes most of the metrics used for image recognition tasks.

Consider the crucial role of performance in APIs through examples like car license plate recognition or police facial recognition systems [11]. In the case of car license plate recognition, an API's ability to quickly and accurately process and analyze images is vital. High-performance APIs can swiftly identify license plates in various conditions, facilitating smoother traffic management and law enforcement. Similarly, in police facial recognition, the speed and precision of an API are critical for identifying suspects or missing persons in real-time, potentially aiding in crime prevention or resolution.

The importance of API performance includes aspects such as:

1. **Latency:** Measures the time an API takes to process an image and generate results. Low latency is essential for real-time applications, where quick response times can be the difference between catching a criminal in the act or missing them entirely.

2. **Accuracy:** High accuracy is crucial, especially in sensitive applications like medical diagnostics or security systems. An API that accurately recognizes faces or medical anomalies can have significant implications for patient care or public safety.

3. **Scalability:** The ability to handle numerous requests simultaneously is vital in high-traffic scenarios, ensuring that the system remains efficient and responsive even under heavy loads.

In conclusion, the performance of APIs in image recognition tasks is not just a technical requirement but a pivotal factor that can influence the efficacy and reliability of crucial applications like traffic management and public safety. The balance between speed, accuracy, scalability, consistency, and resource efficiency defines the utility and success of these technologies in real-world scenarios.

Task	**Metric**	**Description**
Image Classification	Accuracy	Proportion of correct predictions
	Precision	Correct positive predictions relative to total predicted positives
	Recall	Correct positive predictions relative to actual positives
	F1 Score	Weighted average of Precision and Recall
	Confusion Matrix	Table describing model performance
	ROC Curve and AUC	Measure of model's ability to distinguish classes
Image Segmentation	Intersection over Union (IoU)	Overlap between predicted segmentation and ground truth
	Dice Coefficient	Similar to IoU, measures overlap with different formula
	Pixel Accuracy	Ratio of correctly predicted pixels
	Boundary Accuracy	Accuracy of segment boundaries prediction
Object Detection	Average Precision (AP)	Precision at different recall levels, averaged over classes
	Mean Average Precision (mAP)	Mean of AP across all classes
	Intersection over Union (IoU)	Overlap between predicted boxes and ground truth
	Recall	Ratio of correctly detected objects
Pose Extraction	Percentage of Correct Keypoints (PCK)	Accuracy of detected keypoints relative to a threshold
	Average Precision for Keypoints	Similar to mAP but for keypoints
	MSE/RMSE	Mean squared error for regression-based approaches

Table 4.1.: Summary of Metrics for Different Image Recognition Tasks

4.2.2. Cost

The importance of cost in selecting APIs can be exemplified in scenarios like small business e-commerce platforms or budget-constrained systems. For small e-commerce businesses, using a cost-effective API for image recognition can be the difference between a viable, profitable operation and one burdened by excessive overheads. In these cases, an affordable API can efficiently handle product image categorization without compromising much on quality, enabling these businesses to compete in the digital marketplace effectively.

Similarly, in healthcare, especially in underfunded or developing regions, the cost of APIs for medical image analysis can significantly impact their adoption. Budget-friendly APIs can make advanced diagnostic tools accessible to a broader range of healthcare providers, potentially improving patient outcomes in cost-sensitive environments.

Table 4.2 provides a comparative overview of various features and costs associated with different image and video analysis APIs from industrial technology providers, including Google Cloud Vision, Amazon Rekognition, Microsoft Azure Computer Vision API, IBM Watson Visual Recognition, and Clarifai. The table breaks down critical details such as free tier or trial offerings, costs for image and video analysis, and special features each API provides. Additional notes regarding unique billing or service aspects are also included.

Features / API Providers	**Free Tier / Trial**	**Image Analysis Costs**	**Video Analysis Costs**	**Special Features**	**Additional Notes**
Google Cloud Vision	$300 credits for first 90 days. Free monthly usage includes Cloud Vision: 1,000 units/month, AutoML Vision: 40 node hours for training/prediction.	1,001-5,000,000 images: $4.50 per 1,000 images; $0.10 for storage per 1,000 images. 5,000,001-20,000,000 images: $1.80 per 1,000 images; $0.10 for storage per 1,000 images.	AutoML Vision: 40 node hours for training and prediction.	AutoML Vision, Edge training, Batch classification.	AutoML Vision: $3.15 per node hour.

Features / API Providers	**Free Tier / Trial**	**Image Analysis Costs**	**Video Analysis Costs**	**Special Features**	**Additional Notes**
Amazon Rekognition	12-month free tier includes 5,000 images/month for image analysis, 1,000 face vectors storage, 1,000 free minutes of video analysis/month.	Basic label detection: $0.0010/image for first 1M, $0.0008/image for next 1.5M. Image properties: $0.00075/image for first 1M, $0.0006/image for next 1.5M.	Stored video and streaming video events have separate pricing. Free tier includes 1,000 minutes/month.	Face metadata storage, Custom Labels, Face Liveness.	Face metadata storage: $0.00001/face metadata per month. Custom Labels free tier: 10 training hours/month, 4 inference hours/month.
Microsoft Azure Computer Vision API	5,000 free transactions/month.	S1 Tier: $1 per 1,000 transactions for 0-1M transactions, $0.80 per 1,000 transactions for 1M-5M transactions. Rates for S2 and S3 tiers not detailed.	Not detailed.	OCR, Adult content detection, Celebrity and Landmark recognition, etc. Customized Object Detection and Image Classification in preview.	Image Retrieval Text: $0.014 per 1,000 transactions. Customized services in preview with unspecified rates.
IBM Watson Visual Recognition	Not detailed	Not detailed	Not detailed	Not detailed	Not detailed

Features / API Providers	Free Tier / Trial	Image Analysis Costs	Video Analysis Costs	Special Features	Additional Notes
Clarifai	1,000 free operations/-month.	Essential: Starts at $30/month. Professional: Starts at $300/month.	Not detailed	Not detailed	Each paid plan allows users to consume operations at no additional cost up to the package price. Additional operations are charged at the respective unit price per operation.
Imagga	1,000 free API request-s/month.	Indie: $79/month for 70,000 API requests. Pro: $349/month for 300,000 API requests.	Not detailed	Not detailed	Enterprise: Custom built for above 1,000,000 API requests, pay per use.

Table 4.2.: Comparative Overview of Image Recognition API Features and Costs [38, 8, 62, 43, 30, 47]

The pricing structures of the various application programming interfaces (APIs) for image recognition are varied and can be adapted to match particular requirements and patterns of usage.

The Google Cloud Vision API provides a free trial that includes $300 in credits for the first ninety days. This trial includes free monthly usage of 1,000 units for Cloud Vision as well as specific allowances for AutoML Vision. Pricing for custom services is tier-based and determined by the number of features that are included in each 1,000 units, with the first 1,000 units at no cost per month. AutoML Vision has established a price of $3.15 per node hour.

Amazon Rekognition offers various pricing models, including those for face liveness checks, face metadata storage, custom labels, and image and video analysis. The pricing structure for image properties varies. For example, the cost of analyzing 1.5 million photos depends on the pricing structure. A free tier is available for video analysis for the first one thousand minutes each month for a period of twelve months; beyond that, further fees will be charged. The cost of custom labels and face liveness tests is determined by

the application.

In addition to the free tier of 5,000 transactions per month, the Microsoft Azure Computer Vision API offers additional tiers such as S1 for web and container usage, each with a different cost structure for each transaction. On Rapid API, there are a variety of subscription tiers, ranging from Basic to Mega, each providing a different number of requests each month.

Both IBM Watson Visual Recognition and Clarifai offer a variety of pricing levels that vary according to usage and extra operations. Additionally, many operations are free of charge up to a specific minimum limit. Different pricing plans, ranging from free to enterprise, are offered by Imagga, determined by the number of API queries needed.

In a disappointing turn of events, the IBM Watson Visual Recognition API does not have its complete pricing information readily available. However, it is clear that each API provides a variety of pricing alternatives to accommodate a wide range of user requirements and financial constraints.

4.2.3. Functionality

The functionality provided by an API defines its potential to transform raw data into valuable insights and actions. By examining the distinct capabilities of various APIs, developers can align their choice with the specific needs and objectives of their projects.

For instance, an e-commerce platform might benefit from an API that excels in visual search and object detection to enhance the shopping experience by allowing users to search for products using images. In contrast, a security system would require an API with strong facial recognition and motion detection capabilities to identify unauthorized individuals or detect suspicious activities.

In healthcare, the precision and accuracy of image analysis for diagnostic purposes can be a matter of life and death, thus requiring an API that provides high-level medical imaging features. Meanwhile, a social media company might prioritize content moderation features to filter out inappropriate content and protect its user base.

The API's functionality directly affects not only the user experience by providing swift and accurate responses but also has implications for backend processes. It can influence the load on servers, the complexity of integration, and the scalability of the application as user numbers grow.

Table 4.3 does not merely list features; it is a guide to understanding how each API can serve as a tool that molds the user experience. By leveraging the strengths of these APIs, developers can build applications not just for the present needs but also for future scalability and adaptability.

API	**Key Features**	**Strengths**
Google Cloud Vision	Emotion, Object, Text, Motion Detection; Scene Reconstruction; Logo, Explicit Content, Video Detection	For diverse image recognition needs, leveraging Google's advanced ML models for extensive image labeling and OCR.

API	Key Features	Strengths
Amazon Rekognition	Face liveness, Content moderation, Object/Scene/Face detection, Visual search, Face comparison	For robust image/video analysis and facial recognition, excelling in accurate identification and content analysis.
Microsoft Azure Computer Vision API	Image analysis, OCR, Facial recognition, Image tagging	For applications requiring advanced image processing and text extraction, offering responsible facial recognition.
IBM Watson Visual Recognition	Image tagging, Face recognition, Similar image finding, Deep learning classifiers	For high accuracy image analysis using deep learning, suitable for a variety of image recognition tasks with custom model training.
Clarifai	AI workflows, Image/Video recognition, Generative AI, Natural Language Processing	For building AI-powered apps quickly, with a focus on comprehensive AI technologies including NLP and generative AI.
Imagga	Color extraction, Object recognition, Image categorization, Tagging, Facial Recognition	For content-aware cropping and color extraction, offering instant image classification and effective facial recognition in images and videos.

Table 4.3.: Comparative Overview of Key Features and Strengths of Various Image Recognition APIs [40, 6, 64, 41, 28, 45]

Table 4.3 shows that the Google Cloud Vision API is celebrated for its expansive range of functions and advanced machine learning models, making it appropriate for a multitude of image recognition purposes. Conversely, the Microsoft Azure Computer Vision API is recognized for its superior text extraction and conscientious facial recognition features. Amazon Rekognition excels with its facial recognition and detailed content analysis. For applications requiring deep learning in image analysis, IBM Watson emerges as a particularly robust choice. Imagga demonstrates expertise in content-aware image cropping and color extraction, while Clarifai provides an extensive platform designed for developing comprehensive AI-integrated applications.

APIs for cloud-based image analysis come with a wide variety of features and advantages, including emotion detection, object detection, text detection, motion analysis, scene reconstruction, logo detection, and explicit content detection. These functionalities are available through the Google Cloud Vision API. Powerful pre-trained machine learning models, fluid interaction with applications, broad picture labeling, and optical character recognition capabilities are among its features.

Amazon Rekognition offers face liveness, content moderation, object detection, scene detection, face detection, visual search, and comparison of faces. It is renowned for its extensive content analysis capabilities, its excellent image and video analysis capabilities, and its accurate recognition of objects, scenes, and faces.

Image analysis, OCR, facial recognition, and prebuilt image tagging are some of the

functions included in the Microsoft Azure Computer Vision API. It provides advanced methods for image processing, adaptability across a wide range of applications, responsible facial recognition, and efficient text extraction.

Image tagging, face recognition, similar image finding, and deep learning neural net-based classifiers are among the capabilities included in IBM Watson Visual Recognition. By utilizing deep learning methods, it provides high accuracy in picture analysis, training for bespoke classifier models, and performs well for various image recognition applications.

Intelligent workflows, image and video recognition, generative artificial intelligence, and natural language processing are features provided by Clarifai. It is known for its extensive collection of artificial intelligence technologies, support for various programming languages, and high accuracy in building AI-powered applications using photos, videos, text, and audio.

Color extraction, object recognition, image categorization, tagging, and facial recognition are features included in Imagga. It specializes in content-aware cropping, efficient color extraction, instant image categorization, and facial identification in both still images and live stream videos.

4.2.4. Limitations

Table 4.4 serves as a crucial guide in selecting the most appropriate application programming interface (API) for image recognition tasks, considering their inherent limitations. Understanding these limitations is essential, as they can significantly influence the suitability and effectiveness of an API in various practical scenarios. Factors such as file size limits, image dimension constraints, batch processing abilities, and unique features of each API are thoroughly compared in the table.

For instance, an API with a small file size limit might not be ideal for high-resolution satellite imagery or medical imaging applications, where large file sizes are common. On the other hand, APIs with constraints on image dimensions might not be suitable for applications in digital advertising, where a variety of image sizes are required.

Batch processing capabilities are equally important. An API with limited batch processing might not be effective for applications dealing with large volumes of images, such as social media platforms or cloud-based photo storage services. However, for smaller-scale projects or mobile applications, an API with basic batch processing capabilities might be sufficient and more cost-effective.

Specific limitations of an API can impact specialized use cases. For example, an API with limited object recognition capabilities might not be suitable for automated retail checkout systems, which require the identification of a diverse range of products.

Selecting an API based on its limitations is not just about technical compatibility; it's about ensuring that the chosen API aligns with the specific needs of the project and the user experience it aims to provide. The right API can enhance application efficiency, improve user experience, and drive innovation, whereas overlooking these limitations can lead to operational challenges and unmet user expectations. The summary in the table is intended to assist users in making informed decisions by providing a clear understanding of what each API can offer within its limits.

API	Limitation Description
Google Cloud Vision API	Batch processing limited to 8MB per request
Amazon Rekognition	Max image size: 15 MB (S3 object), 5 MB (raw bytes) Max image dimensions: 10K x 10K pixels Face detection minimum: 40x40 pixels in 1920x1080 Max face vectors in a collection: 20 million Video support: Up to 10GB, 6 hours length
Microsoft Azure Computer Vision	Max file size: 4 MB (API v3.2), 20 MB (API v4.0) TPS limit (Free Tier): 20 transactions/min
IBM Watson Visual Recognition	Max image size: 10 MB (images/zip files)
Clarifai API	Supports POST, PATCH, GET, DELETE requests
Imagga API	Best practices suggest images not larger than 300px on the shortest side Provides specific HTTP response codes for various errors

Table 4.4.: Comparison of Limitations for Various Image Recognition APIs [39, 7, 63, 42, 29, 46]

The Google Cloud Vision API supports various web image formats such as GIF, BMP, WebP, Raw, ICO, and others. The maximum file size is not expressly specified; however, batch processing support is restricted to a maximum of 8MB per request. The platform lacks built-in video capabilities and only provides a single HTTP endpoint for API requests. The API provides functionality for object detection, face detection, sentiment detection (faces), OCR, and other features. However, it encounters challenges with the accuracy of face detection in specific situations, and the capability of sentiment detection is restricted to only four fundamental emotions. The pricing structure is somewhat higher compared to Amazon Rekognition.

Amazon Rekognition is compatible with JPG and PNG file formats. The maximum size for a picture is 15 MB when it is saved as an Amazon S3 object. The picture dimensions required for various detections have different maximum limits, with face detection necessitating a minimum size that depends on the image dimensions. The detectText function can identify and analyze a maximum of 100 words within an image. The video analysis feature can be used for videos stored and have a maximum size of 10GB and a duration of up to 6 hours. Nevertheless, it encounters challenges with detecting faces in black and white photographs and among older individuals. The emotion recognition capabilities of this system are more extensive compared to Google Cloud Vision.

The free tier of the Microsoft Azure Computer Vision API has a limit of 20 transactions per minute, while the S1 tier allows for up to 30 transactions per second. The maximum allowable file size is 4 MB for API version 3.2 and 20 MB for version 4.0. The API does not have the capability to deliver numerous photos in a single call. However, it supports

Image Analysis and OCR in many languages. On-premises implementation includes the availability of OCR functionality.

IBM Watson Visual Recognition was discontinued on December 1, 2021. The maximum picture size was 10 MB, and it was compatible with image formats such as .gif, .jpg, .png, and .tif. For URL picture analysis, it is advisable to have a minimum pixel density of 32×32 pixels.

The Clarifai API facilitates software-to-software communication with clients such as Python, Node, and Java. The system supports a range of request types, such as POST, PATCH, GET, and DELETE. The retrieved literature does not provide specific information regarding restrictions on file size, image dimensions, batch processing, or other technical limits.

Imagga's documentation offers recommendations on picture size, advising that photos should not exceed 300px on the shortest side for optimal results. The API provides error management by utilizing several HTTP response codes to indicate different types of problems, like "400 Bad Request", "401 Unauthorized", "429 Too Many Requests", and others. The literature also includes guidelines for optimizing image processing techniques and establishing confidence thresholds to ensure accurate tagging. The documentation did not provide precise information regarding specific numerical constraints such as file size, image dimensions, and batch processing capabilities.

4.3. Methodology Summary

APIs have a crucial role in multiple industries, such as e-commerce, security, and healthcare, especially in the field of image recognition. Their evaluation is based on crucial elements such as latency, accuracy, consistency, scalability, and resource consumption. Python programs provide an efficient method to analyze and utilize multiple APIs, including Imagga, Amazon Rekognition, IBM Watson Visual Recognition, Microsoft Azure Computer Vision, and Clarifai's API. A Python module has been developed to optimize the process of extracting textual data from image recognition APIs. This module consists of a Python program for retrieving documentation, together with an advanced module that provides support for querying and vectorization functionalities.

4.3.1. Data Collection Pipeline

The process of web page analysis involves several key steps to ensure efficient and accurate data retrieval. Initially, the pipeline undertakes destination checking to ascertain the suitability of the webpage for further analysis. This entails verifying if the webpage has been previously visited and if it is not intended for file downloads. Following this, the method proceeds to initiate a webpage visit, analogous to knocking on a door to gauge accessibility. If the webpage is reachable, the method progresses; otherwise, it promptly notifies of any issues encountered. Subsequently, upon gaining access, the method delves into page reading, meticulously scanning the webpage's content, particularly focusing on paragraphs of text. This approach mirrors the process of scanning through a book for

specific paragraphs of interest, laying the groundwork for comprehensive data extraction and analysis.

1. **Destination Checking:** Initially, the method verifies whether it has previously visited the webpage and whether the webpage is not designated for file downloads. If it has visited the page before or if it's a download page, it takes no further action.
2. **Webpage Visit:** Subsequently, it attempts to access the webpage, akin to knocking on a door and awaiting a response. If the webpage is reachable, it proceeds; otherwise, it halts and reports an issue.
3. **Page Reading:** Upon gaining access, the method commences reading the content of the webpage, with particular attention to paragraphs of text. This process resembles scanning through a book for specific paragraphs.

Challenges and Limitations

Challenges include downloading links, extracting text from multimedia, bypassing scraper blocks, and processing non-Latin languages.

1. Constraint when the link leads to a downloadable file.
2. Incapacity to extract textual content from videos or other multimedia formats.
3. Obstacles in accessing websites that actively block web scrapers.
4. Difficulties encountered with languages beyond ASCII or English, including UTF-8 encoded languages such as Chinese or Russian.

4.3.2. Representation of Data

This section outlines the methodology for transforming and analyzing textual data, specifically focusing on the vectorization and similarity analysis of words within a sentence.

Vectorization and Similarity Analysis

Preparation: This process starts with the cleaning from any special characters of both the word and the sentence, ensuring they are in a suitable format (UTF-8) for analysis.

Transformation: The next step involves converting the word and each word in the sentence into a format known as a 'vector', by translating them into distinct patterns or codes that are understandable by the machine.

Pattern Comparison: The method is places side by side the pattern of the individual word with the patterns of each word in the sentence.

Similarity Computation: The process then calculates the degree of similarity between the individual word and the words in the sentence. This can be done using various methods such as averaging (mean), identifying the highest match (max), or finding the median value of these similarities.

1. **Mean:** This method provides an overall assessment of how well a given word fits within the context of a sentence. It does this by averaging the similarity scores between the given word and each word in the sentence. For example, if the given word is "price" and the sentence is "This domain costs 200 dollars," the words "costs" and "dollars" are similar to "price." By calculating the average similarity of "price" with all the words in the sentence, the method determines the overall contextual relevance of "price" to the sentence. Thus, the higher the average similarity, the more significant the given word is within the context of the sentence.

2. **Median:** This method assesses how well a given word fits within the context of a sentence by determining the median similarity score between the given word and each word in the sentence. For example, if the given word is "price" and the sentence is "The API offers a variety of features at different costs, including discounts on several products" the words "costs" and "discounts" are similar to "price." By calculating the median similarity of "price" with all the words in the sentence, the method identifies the middle value of these similarities.

   Using the median instead of the mean reduces the impact of outliers or extremely high or low similarity scores. For instance, if the sentence contains a word that is entirely unrelated to "price" and has a very low similarity score, it would significantly lower the mean similarity score. However, it would not affect the median as much. This means that the median provides a more stable and representative measure of the typical relevance of the given word within the sentence, ensuring that the overall assessment is not unduly influenced by any single word's similarity score.

3. **Maximum:** This method assesses how well a given word fits within the context of a sentence by determining the maximum similarity score between the given word and each word in the sentence. For example.

   Using the max instead of the median or mean highlights the strongest single relationship between the given word and any word in the sentence. For instance, if "costs" has a very high similarity score with "price," this score would be chosen as the representative value. This means that the max method select the most significant connection.

Each of these methodologies provides distinct insights into how the word correlates with the sentence.

Decision-making: Based on these computations, the method determines whether the word exhibits sufficient similarity to the sentence.

4.4. Categorization of Data

Following the intricate process of converting sentences into vector representations and evaluating their similarity against predefined categories such as "performance", "cost", "limit" and "functionality" the method proceeds to meticulously curate the highest-scoring

sentences for retention and analysis. Each sentence undergoes a meticulous pairing with its corresponding similarity score, thereby crafting a comprehensive key-value pair within a structured dictionary. This dictionary encapsulates the essence of the analyzed sentences, with each sentence serving as a key and its respective similarity score as its corresponding value.

Subsequently, this meticulously structured data undergoes systematic organization into distinct JSON files, aptly named to reflect the essence of the category to which the sentences belong. For instance, sentences bearing the closest affinity to the "performance" category find their archival sanctuary within a designated file christened "performance.json" within these meticulously crafted files, the sentences unfold in a meticulously orchestrated symphony, arranged in a descending order of their scores. This deliberate arrangement ensures that the sentences revealing the highest levels of similarity are accorded prominence and visibility within the file hierarchy.

Moreover, the method goes a step further in its meticulous curation process by prioritizing the inclusion of the top three sentences from each category into the JSON files. This strategic selection mechanism ensures that the stored content resonates with precision and authenticity, reflecting the utmost levels of similarity to the respective category. Through this meticulous selection and organization process, the method not only facilitates efficient data storage but also empowers users with insights derived from the highest-scoring sentences.

4.5. Graphical Representation

The directory structure for the data is carefully organized into a multi-tiered hierarchy, as depicted in the illustrations. The apex of this structure is demarcated by root directories, each labeled to correspond with a specific type of similarity measure: "mean", "median", and "max". These categories serve as the primary organizational units.

Descending into this structure, we encounter subdirectories, systematically named for each API provider under evaluation: Google Cloud Vision API, Amazon Rekognition, Microsoft Azure Computer Vision API, Clarifai, and Imagga.

Delving further into the subdirectories assigned to each API, we find a quartet of JSON files. These files are purposefully segregated to represent distinct facets of the API's features: "performance", "cost", "limit", and "functionality". Such a structured layout of directories and files creates an intuitive and navigable framework, simplifying the process of accessing and comparing the different facets of API features.

This organized approach is designed for optimal clarity and user-friendliness, ensuring that data pertaining to each API's feature set can be easily located, accessed, and compared, thus streamlining the user experience in data analysis and retrieval tasks.


```
{
"Fast-track data analysis and automate data insights": 0.435,
"Enhance data analytics for better marketing, forecasting, and insights.": 0.419,
"Create actionable insights through data intelligence.": 0.4099,
"3. Supporting cloud-native application development": 0.403,
"New Apigee API Monitoring Metrics": 0.36330172419548035,
"Voice user interface in devices and applications": 0.3473,
```

```
"Discover data solutions for startups": 0.346,
.
.
.
}
```

Listing 4.1: A JSON code snippet showing how data is saved.

In Listing 4.1, the provided text showcases a snippet from a JSON file, a repository for structured data. It delineates costs linked to Google API services, each accompanied by a numerical score denoting similarity to the "cost" category. A more detailed example of an analysis result for the Amazon Rekognition API looking only at the functionality (according to the default terms (see Appendix A.15)) and using the Maximum-method can be found in the Appendix A.14.

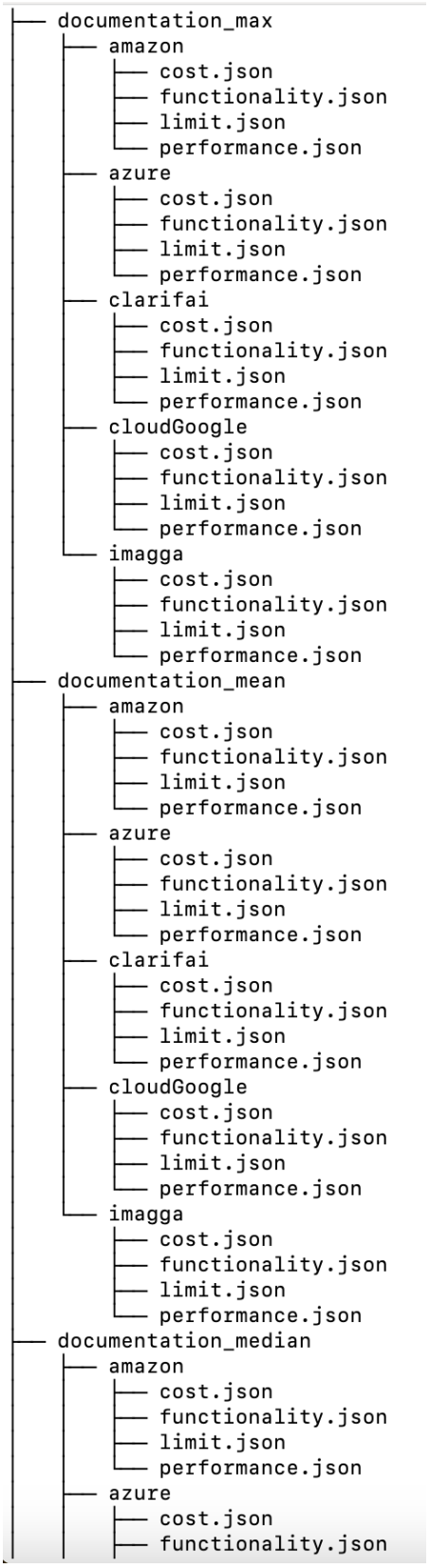

Figure 4.1.: Hierarchical Structure of API Documentation: Organized by Max, Mean, and Median

4.6. User Interface and Features

In this section, we delve into the user interface and the functionality of comparing support tool designed for the analysis and comparison of API documentation. The goal is to dissect the operational flow and highlight the interactive elements that users will engage with, from initial data input to the final analysis output.

4.6.1. Operational Flow Diagram

The operational flow diagram acts as a visual representation of the comparing support tool architecture, mapping out the user journey and the application's response to each interaction. It is an essential asset for users who aim to navigate the complexities of API documentation analysis, offering a step-by-step guide through the support tool functionality.

As illustrated in Figure 4.2, the operational workflow encapsulates the systematic progression of tasks the support tool performs. The journey begins when a user initiates the tool, which activates a structured flow of operations.

If the user opts to update the data, the tool embarks on a comprehensive data refresh protocol. This includes loading a pretrained Facebook/FastText vectors model, scraping the latest web pages for new data, and refining this information to ensure quality and relevance. The refined data is then vectorized, leading to a comparative analysis based on maximum, mean, and median similarity scores. The outcome of this analysis is meticulously sorted and stored in four JSON files, each dedicated to a specific categories set: performance, cost, limit, and functionality.

Start App

This is the initial step where the application is launched.

Select the Required Method

The application prompts the user to select one of three methods: Max, Median, or Mean. These methods determine how the data will be processed and analyzed.

Re-scrape Data?

The user is asked if they want to re-scrape the data. If the user selects 'no', the application skipping the re-scrape process. If the user selects 'yes', the following steps are initiated:

Input `max_level`

Purpose: This variable defines the depth of the web scraping process. The scraper starts by fetching all links on the initial page (level 1). It then follows these links to fetch additional links (level 2), and continues this process to the depth specified by `max_level`.

Detail: If `max_level` is set to 3, the scraper will fetch links from the initial page, follow those links to another page, and then follow the links on the subsequent pages.

Input `level_size`

Purpose: This variable specifies the number of links the scraper can process at each level. Given that a single page may contain thousands of links, `level_size` helps manage the number of links processed to avoid overloading the system.
Detail: If `level_size` is set to 100, the scraper will process up to 100 links per level before moving to the next depth level.

Input `threshold`

Purpose: This variable defines the similarity threshold. It sets the minimum similarity score required to consider two pieces of data as similar.
Detail: For example, if the threshold is set to 0.8, only pairs of data with a similarity score of 0.8 or higher will be considered similar.

Input Custom Keywords

Purpose: The user can input their own keywords for the similarity search. These custom keywords will be used to find matches in the documentation.
Detail: If the user does not want to use custom keywords, they can enter 'No' or 'n' to use the default predefined keywords.

Start Scrape Recursively

If the user opts to re-scrape the data, the scraper begins the recursive process based on the specified `max_level` and `level_size`.

Clean and Filter Scraped Data

The scraped data is then cleaned and filtered to remove irrelevant information and improve accuracy.

Convert Sentences to Vectors Using the FastText Module

The application uses the FastText module to convert sentences into numerical vectors, which can then be compared for similarity.

Calculate Similarity Between the Category Keywords and Sentences

The application calculates the similarity between the user-defined (or default) keywords and the sentences in the documentation based on the vectors generated by the FastText module.

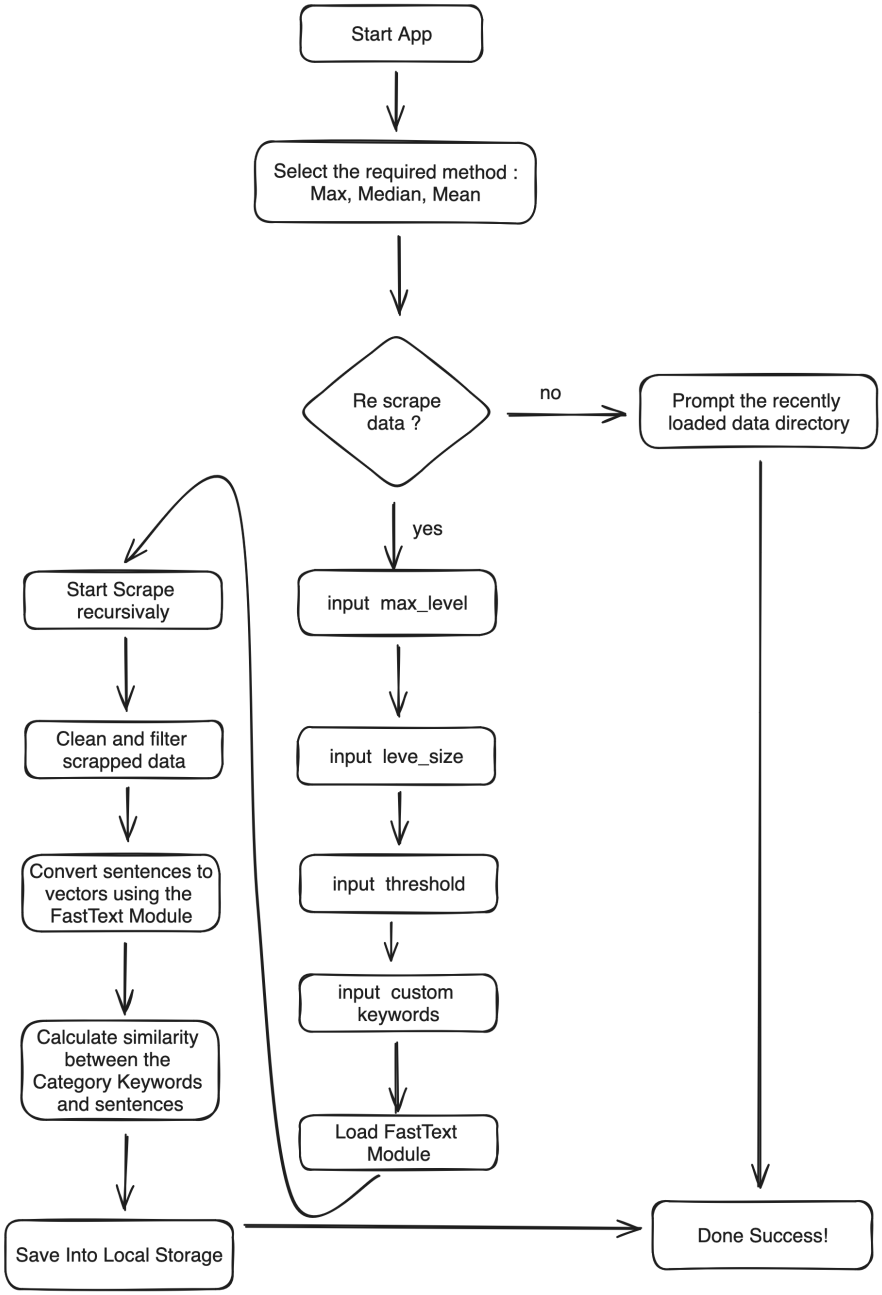

Figure 4.2.: This flowchart outlines the operational steps for the comparing support tool, from start-up to displaying results. It navigates through decision points like reloading data, processing through feature selection, data scraping, cleaning, model utilization, similarity scoring, and storage updating.

4.7. Demo

Upon launching the tool, the user is prompted to select the desired method from three options: median, mean, and max, as shown in Figure 4.3, which have been previously explained in Section 4.3.2

```
Select Method Enter your choice (e.g., 1 or 2 ..)
 1- median
 2- mean
 3- max
 2
```

Figure 4.3.: Option for User to Select the needed Method

The next step is to refresh the data, as shown in Figure 4.4.

```
Do you want to reload the documentation ? (yes/no):
```

Figure 4.4.: Prompt to Choose Whether to Reload Data

If the user selects 'yes' or 'y', the tool initiates an extensive data refresh procedure. Due to the comprehensive nature of this task, it may take over two hours to complete. Once finished, the application updates the local storage and associated JSON files with the newly acquired data. Alternatively, selecting 'no' or 'n' will ask the user to check with the existing data stored locally, as shown in Figure 4.5.

```
Do you want to reload the documentation ? (yes/no): n
Please check the most recently loaded data in the JSON files located in the current working directory:
'/Users/ziadelsarrih/DesktopMP/trunk/master/_doc_method_3'
```

Figure 4.5.: Prompt Showing the 'No' Option Selected for Reloading Data

Once the user decides to reload the data and selects 'yes' or 'y', the application will display three prompt messages. The first prompt asks for the **deep level** value, the second for the **level size**, and the third for the **Similarity Threshold** as shown in Figure 4.6.

```
Enter deep level required from 1 to 10000: 2
Enter level size required from 1 to 10000: 10
Enter Similarity Threshold that consider is similar between 0 and 1: 0
```

Figure 4.6.: Prompts for entering the deep level, level size, and Similarity Threshold after selecting 'yes' to reload data

The next step is to provide the user with the option to use custom keywords for finding similarities. This means that the user can input their own keywords, which the application will use to search for matches or related terms in the documentation. If the user prefers to use the default set of predefined keywords, they can simply enter 'No' or 'n'. This flexibility allows for a more tailored and relevant search experience based on the user's specific needs as shown in Figure 4.7.

```
Do you want to enter the keywords ? (yes/no): y
Enter new keywords for 'cost' separated by commas (or press Enter to skip):
price, expense, subscription, payment, pay
'cost' category keywords is been updated:
Enter new keywords for 'limit' separated by commas (or press Enter to skip):
boundary, threshold, restriction
'limit' category keywords is been updated:
Enter new keywords for 'performance' separated by commas (or press Enter to skip):
throughput, efficiency, speed, score
'performance' category keywords is been updated:
Enter new keywords for 'functionality' separated by commas (or press Enter to skip):
feature, functionality, facial recognition, OCR, analytics
'functionality' category keywords is been updated:
```

Figure 4.7.: Prompt allowing the user to input custom keywords or select the default predefined ones for similarity search

4.8. Summary

This section discusses the methodology used for data collection and analysis. It includes web page analysis for data retrieval, limitations in data extraction (such as handling non-Latin languages), and the representation of data through vectorization and similarity analysis. The categorization of data involves sorting similarity-scored sentences into JSON files under specific categories like performance, cost, limitations, and functionality. The graphical representation of API documentation is organized by similarity measures (mean, median, max) and API providers. A user interface and operational flow diagram are provided for the comparison support tool, illustrating the process from data input

to final output. In summary, this thesis offers a detailed and systematic evaluation of major image recognition cloud services. It assists users in comparing the APIs according to their specific needs, based on a thorough understanding of each service's performance, cost, functionality, and limitations.

5. Summary, Conclusion, and Recommendations

The rapid advancements in computer vision technologies, particularly in the development of robust image recognition APIs by industrial giants like Amazon, Google, IBM, and Microsoft, have been a game changer for programmers and businesses. These APIs have revolutionized the application of computer vision by offering powerful AI capabilities and eliminating the need to build complex systems from scratch. However, they also present significant challenges in terms of performance, cost, limitations, and functionalities. This thesis provides an in-depth analysis and comparison of these APIs to assist developers and organizations in making well-informed choices, emphasizing the importance of selecting the most suitable API for specific requirements.

The introduction of the thesis explores the history of AI, focusing on machine learning and deep learning. It recalls how research in AI began in the mid-20th century, initially showing slow progress, especially in fields like voice and sight recognition. The early challenges faced in replicating intuitive human tasks such as visual and auditory identification in machines are discussed. The narrative shifts around 2010 with the implementation of neural networks, leading to a breakthrough in AI and machine learning. The resurgence of neural networks, inspired by the human brain, marked a new era in technology, particularly in deep learning. Since then, there has been a surge in research and development, with deep learning making significant strides in computer vision, natural language processing, speech recognition, and more. The integration of deep learning in everyday devices like smartphones and the growing potential in autonomous vehicles are highlighted as examples of its impact.

The thesis then shifts focus to image recognition. It discusses the use of image recognition APIs, which allow developers to incorporate pretrained models into applications. The significance of these APIs in various sectors is underscored, emphasizing their role in simplifying complex image recognition tasks. The thesis covers a range of APIs, including Amazon, Google, IBM, and Microsoft, showcasing their abilities in facial analysis and object recognition. The necessity for a framework to compare these APIs is established, outlining the study's aim to evaluate their performance, cost, functionality, and limitations.

Chapter one dives into the background of computer vision APIs, shedding light on their contributions. It discusses the diverse features they offer, such as text reading, handwriting analysis, and face recognition. The chapter emphasizes the convenience of these cloud-based APIs and their role in democratizing access to AI capabilities. The importance of evaluating and contrasting these APIs is stressed, considering accuracy, cost, and specific project requirements.

In chapter two, the thesis provides an overview of several key APIs. This chapter

emphasizes the need for software quality analysis and data-driven insights for assessing these APIs. It explores recent developments in face recognition research, focusing on CNNs and the challenges of biases in datasets. The chapter underscores the importance of overcoming bias in AI systems to maintain fairness and uphold ethical standards.

The research aim and questions section sets clear objectives to examine and compare image recognition APIs, focusing on four categories: performance, cost, functionality, and limitations. It proposes the development of a comparison support tool to aid in comparing APIs based on these categories. The research questions delve into aspects such as accuracy issues, API performance, cost implications, functionality level, and limitations.

The methodology section outlines the approach for data collection and analysis, including web scraping libraries and data processing strategies. It discusses the use of the FastText model for text classification and word representation, tokenization processes, and clustering analysis. The section concludes with an explanation of the decision support tool's operational flow, from data input to final output.

In the results and discussion section, the thesis presents the findings of the research. It provides a detailed examination of various image recognition APIs across all categories. Graphical representations of API documentation, categorized by similarity measures and providers, are included. The decision support tool's user interface and functionality are also discussed, demonstrating its utility in guiding users to select the most suitable APIs.

In conclusion, this thesis offers a comprehensive analysis of image recognition APIs, contributing to a deeper understanding and facilitating informed decision-making for developers and organizations. The development of the decision support tool, leveraging machine learning for analysis, simplifies the API selection process and aligns capabilities with user-specific requirements. The study's findings enhance the knowledge of current API offerings and provide practical tools to aid in the effective and reliable development of applications in the field of computer vision.

6. Future Work

6.1. Advancements in Chatbot Development: Open-Source Module Implementation

Building upon the foundational clustering study, this research now turns towards the next critical phase in chatbot development: the creation of a chatbot model using open-source chatbot modules. These modules come in various forms, each offering unique advantages and functionalities. This exploration will delve into prominent chatbot modules like Rasa, ChatterBot, and Dialogflow, examining their capacity for training chatbots, framework suitability for specific projects, and capability to generate user-centric responses.

The initial step in this endeavor involves selecting an open-source chatbot framework that best aligns with the project's objectives. Critical factors influencing this choice include dataset size, desired chatbot features, and budget constraints. This analysis will review several renowned frameworks employed in chatbot development, assessing their suitability for different project requirements.

Rasa, an acclaimed open-source chatbot framework, is notable for its proficient use of machine learning in chatbot development. It stands out for its flexibility and scalability, making it an ideal choice for handling large datasets and high-traffic scenarios [59]. ChatterBot, on the other hand, is celebrated for its simplicity and practicality, utilizing a rule-based approach that caters to a wide spectrum of applications [4]. It offers an approachable solution for projects prioritizing ease of use and rapid deployment. Dialogflow, developed by Google, leverages machine learning to provide a user-friendly platform capable of handling diverse natural language queries [69]. It is particularly advantageous when the project demands sophisticated natural language processing capabilities.

The selection of an open-source chatbot framework must be guided by the project's specific needs and goals, considering aspects like dataset complexity, required features, and ease of integration.

6.1.1. Training the Chatbot

Following the framework selection, the next crucial step is the chatbot's training. This process enables the chatbot to acquire the necessary skills to accurately comprehend and respond to human inquiries. Training involves exposing the chatbot to a dataset encompassing user interactions and cluster attributes. These interactions form the foundation of the chatbot's learning curve.

Manual curation of training data through dialogues and leveraging tools like Google Search Console to capture real user interactions are pivotal in shaping the chatbot's

comprehension abilities. Machine learning algorithms are instrumental in linking user needs with specific cluster characteristics, enabling the chatbot to associate user queries with relevant responses.

6.1.2. Developing Responses

Post-training, the focus shifts to crafting responses that the chatbot will deliver. These responses should be tailored to align with user needs and the unique attributes of each cluster. Chatbot developers can manually create responses for varied inquiries or employ automated tools like Rasa NLU for real-time response generation. The objective is to ensure that the chatbot's responses are relevant, accurate, and user-friendly, thus enhancing the overall user interaction experience.

For instance, when a user inquires about face recognition APIs, the chatbot should be able to provide information relevant to the "Face Recognition" cluster. This ability to provide targeted, valuable information is critical in elevating the user experience with the chatbot.

6.2. Model Training and Testing

The development of a chatbot does not end with its activation; rather, it marks the beginning of a crucial phase of training and evaluation. This section focuses on the partitioning of data for training and testing, the training process, and the evaluation of the chatbot model's performance.

6.2.1. Splitting the Data

The first step involves dividing the dataset into a training set and a testing set. The training set forms the basis of the chatbot's learning, while the testing set serves as a means to objectively evaluate its performance. A common approach is to allocate 70% of the data for training and the remaining 30% for testing, although these proportions can be adjusted based on specific project needs.

6.2.2. Training the Chatbot Model

With the data appropriately divided, the next step is to train the chatbot model using machine learning techniques. This involves employing approaches like supervised learning for training with annotated data, unsupervised learning for pattern identification, and reinforcement learning for adaptive learning through trial and error.

6.2.3. Evaluating Performance

Evaluating the chatbot's performance post-training is essential to ascertain its effectiveness in providing accurate API recommendations. Metrics like accuracy and effectiveness are used to measure the precision of the chatbot's responses and user satisfaction levels.

Improving performance may involve optimizing machine learning algorithm parameters, data augmentation, or reevaluating the chosen algorithm.

6.3. Summary

Chapter three of this research provides a comprehensive strategy for enhancing the efficiency of Image Recognition APIs within a chatbot framework. It outlines a systematic approach for data acquisition, modification, and analysis, employing techniques such as vectorization, clustering analysis, and the development of a robust chatbot model. The overarching goal is to significantly enhance

the efficacy of Image Recognition in chatbots, thus contributing to the practical deployment of AI-driven chatbots in various applications. This chapter lays the groundwork for subsequent chapters, which will explore case studies and practical implementations, demonstrating the advantages of this comprehensive approach in enhancing image recognition capabilities in chatbot systems.

Bibliography

[1] A. Abodayeh, R. Hejazi, W. Najjar, L. Shihadeh, and R. Latif. Web scraping for data analytics: A BeautifulSoup implementation. In *2023 Sixth International Conference of Women in Data Science at Prince Sultan University (WiDS PSU)*. IEEE, Mar. 2023.

[2] W. S. Agras and B. G. Kirkley. Bulimia: theories of etiology. In *Handbook of Eating Disorders*, pages 367–378. Elsevier Inc., 1986.

[3] H. Alaskar and T. Saba. Machine learning and deep learning: A comparative review. In K. K. Singh Mer, V. B. Semwal, V. Bijalwan, and R. G. Crespo, editors, *Proceedings of Integrated Intelligence Enable Networks and Computing*, pages 143–150, Singapore, 2021. Springer Singapore. ISBN 978-981-33-6307-6.

[4] N. Albayrak, A. Ozdemir, and E. Zeydan. An overview of artificial intelligence based chatbots and an example chatbot application. In *2018 26th Signal Processing and Communications Applications Conference (SIU)*. IEEE, May 2018.

[5] O. Alvarado and A. Waern. Towards algorithmic experience: initial efforts for social media contexts. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI 2018)*, pages 286–298. ACM, New York, 2018.

[6] Amazon. Amazon rekognition – features. `https://aws.amazon.com/rekognition`, 2024. Accessed: 2024-02-10.

[7] Amazon. Amazon rekognition – limits. `https://docs.aws.amazon.com/rekognition/latest/dg/limits.html`, 2024. Accessed: 2024-02-10.

[8] Amazon. Amazon rekognition – pricing. `https://aws.amazon.com/rekognition/pricing/`, 2024. Accessed: 2024-02-10.

[9] A. Aron, G. W. Lewandowski, D. Mashek, and E. N. Aron. *The self-expansion model of motivation and cognition in close relationships*. Oxford University Press, Apr. 2013.

[10] A. Batlaw. IBM watson. `https://www.ibm.com/watson`, May 2023. Accessed: 2023-9-12.

[11] S. G. Benzell and M. W. Van Alstyne. The role of APIs in firm performance. *SSRN Electron. J.*, 2016.

[12] M. Bertacchi, I. Silveira, and N. Omar. A comparative analysis of the evolution of the IBM watson's visual recognition API on android. In *2017 Workshop of Computer Vision (WVC)*. IEEE, Oct. 2017.

[13] J. Birnholtz, C. Fitzpatrick, M. Handel, and J. R. Brubaker. Identity, identification and identifiability: the language of self-presentation on a location-based mobile dating app. In *Proceedings of the 16th International Conference on Human-Computer Interaction with Mobile Devices and Services*, pages 3–12. Association for Computing Machinery, New York, NY, United States, 2014.

[14] E. Bisong. Google AutoML: Cloud vision. In *Building Machine Learning and Deep Learning Models on Google Cloud Platform*, pages 581–598. Apress, Berkeley, CA, 2019.

[15] G. Boesch. Image recognition: The basics and use cases (2023 guide). `https://viso.ai/computer-vision/image-recognition/`, Jan. 2023. Accessed: 2023-9-12.

[16] A. Bouganis and M. Shanahan. Flexible object recognition in cluttered scenes using relative point distribution models. In *2008 19th International Conference on Pattern Recognition*. IEEE, Dec. 2008.

[17] R. J. Brand, A. Bonatsos, R. D'Orazio, and H. DeShong. What is beautiful is good, even online: Correlations between photo attractiveness and text attractiveness in men's online dating profiles. *Comput. Human Behav.*, 28(1):166–170, Jan. 2012.

[18] J. D. Brown. Mass media influences on sexuality. *J. Sex Res.*, 39(1):42–45, Feb. 2002.

[19] K. D. Brownell. Personal responsibility and control over our bodies: when expectation exceeds reality. *Health Psychol.*, 10(5):303–310, 1991.

[20] J. Buolamwini and T. Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In S. A. Friedler and C. Wilson, editors, *Proceedings of the 1st Conference on Fairness, Accountability and Transparency*, volume 81 of *Proceedings of Machine Learning Research*, pages 77–91. PMLR, 23–24 Feb 2018. URL `https://proceedings.mlr.press/v81/buolamwini18a.html`.

[21] J. Burrell. How the machine "thinks": understanding opacity in machine learning algorithms. *Big Data Soc*, 3:1–12, 2016.

[22] D. M. Buss. Sex differences in human mate preferences: Evolutionary hypotheses tested in 37 cultures. *Behav. Brain Sci.*, 12(1):1–14, Mar. 1989.

[23] D. M. Buss. *The Evolution of Desire*. Cambridge University Press, Sept. 1994.

[24] D. M. Buss. *Evolutionary social psychology*. Cambridge University Press, Sept. 1998.

[25] J. Carr. Big data analytics for cloud, iot and cognitive computing. SlidePlayer, 2018. URL `https://slideplayer.com/slide/14477640/`. Slide 69, 71, 73.

[26] S. Chen, S. Saiki, and M. Nakamura. Toward flexible and efficient home context sensing: Capability evaluation and verification of image-based cognitive APIs. *Sensors (Basel)*, 20(5):1442, Mar. 2020.

[27] T. H. H. Chua and L. Chang. Follow me and like my beautiful selfies: Singapore teenage girls' engagement in self-presentation and peer comparison on social media. *Comput. Human Behav.*, 55:190–197, Feb. 2020.

[28] Clarifai. Clarifai api guide - api overview. `https://docs.clarifai.com/api-guide/api-overview`, 2024. Accessed: 2024-02-10.

[29] Clarifai. Clarifai api guide - api overview. `https://docs.clarifai.com/api-guide/api-overview`, 2024. Accessed: 2024-02-10.

[30] Clarifai. Clarifai pricing. `https://www.clarifai.com/pricing`, 2024. Accessed: 2024-02-10.

[31] M. Cunningham. *Measuring the physical in physical attractiveness*. Cambridge University Press, 1998.

[32] G. V. Cybenko. Linear algebra and learning from data [bookshelf]. *IEEE Control Systems*, 40:71–72, 2020. URL `https://api.semanticscholar.org/CorpusID:218833658`.

[33] R. Das, S. Thepade, S. Bhattacharya, and S. Ghosh. Retrieval architecture with classified query for content based image recognition. *Appl. Comput. Intell. Soft Comput.*, 2016:1–9, 2016.

[34] Y. K. Dwivedi, L. Hughes, E. Ismagilova, G. Aarts, C. Coombs, T. Crick, Y. Duan, R. Dwivedi, J. Edwards, A. Eirug, V. Galanos, P. V. Ilavarasan, M. Janssen, P. Jones, A. K. Kar, H. Kizgin, B. Kronemann, B. Lal, B. Lucini, R. Medaglia, K. Le Meunier-FitzHugh, L. C. Le Meunier-FitzHugh, S. Misra, E. Mogaji, S. K. Sharma, J. B. Singh, V. Raghavan, R. Raman, N. P. Rana, S. Samothrakis, J. Spencer, K. Tamilmani, A. Tubadji, P. Walton, and M. D. Williams. Artificial intelligence (AI): Multidisciplinary perspectives on emerging challenges, opportunities, and agenda for research, practice and policy. *Int. J. Inf. Manage.*, 57(101994):101994, Apr. 2021.

[35] X. Fei, Y. Xie, S. Tang, and J. Hu. Identifying click-requests for the network-side through traffic behavior. *J. Netw. Comput. Appl.*, 173(102872):102872, Jan. 2021.

[36] M. Feinberg. Automatic image tagging and categorization with imagga. `https://cloudinary.com/blog/automatic_image_tagging_and_categorization_with_imagga`, 2015. Accessed: 2024-01-26.

[37] B. Fruchard, S. Malacria, G. Casiez, and S. Huot. User preference and performance using tagging and browsing for image labeling. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, New York, NY, USA, Apr. 2023. ACM.

[38] Google. Cloud vision software - 2024 reviews, pricing and demo. https://www.softwareadvice.com/visual-search/cloud-vision-profile/, 2024. Accessed: 2024-02-10.

[39] Google. Google vision vs amazon rekognition: a comparison - cloud academy blog. https://cloudacademy.com/blog/google-vision-amazon-rekognition-comparison/, 2024. Accessed: 2024-02-10.

[40] Google. Google cloud vision api features - g2. https://www.g2.com/products/google-cloud-vision-api/features, 2024. Accessed: 2024-02-10.

[41] IBM. Ibm research - visual ai lab. https://researcher.draco.res.ibm.com/researcher/view_group.php?id=8269, 2024. Accessed: 2024-02-10.

[42] IBM. Ibm watson visual recognition v3 - developer cloud documentation. https://watson-developer-cloud.github.io/doc-tutorial-downloads/visual-recognition/VisualRecognitionV3.html, 2024. Accessed: 2024-02-10.

[43] IBM. Ibm watson visual recognition pricing, alternatives and more 2024 - trustradius. https://www.trustradius.com/products/ibm-watson-visual-recognition/pricing, 2024. Accessed: 2024-02-10.

[44] K. Ilya. ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[45] Imagga. Imagga api documentation. https://docs.imagga.com, 2024. Accessed: 2024-02-10.

[46] Imagga. Imagga api documentation. https://docs.imagga.com, 2024. Accessed: 2024-02-10.

[47] Imagga. Imagga pricing. https://imagga.com/pricing, 2024. Accessed: 2024-02-10.

[48] Imagga Technologies. Imagga among the top performers in recent comparison of image recognition apis. https://imagga.com/blog/imagga-among-top-performers-image-recognition-apis-comparison/, 2019. Accessed: 2024-01-26.

[49] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov. Bag of tricks for efficient text classification, 2016.

[50] D. Justin. Abhishek, b., and latika. *What is Image Recognition*, 60(6):84–90, 2019.

[51] M. C. Kai Hwang. *Big-Data Analytics for Cloud, IoT and Cognitive Computing*. Wiley, Year: 2017, 2017. ISBN 1119247020,9781119247029. URL https://slideplayer.com/slide/14477640/.

[52] A. Khalil, S. G. Ahmed, A. M. Khattak, and N. Al-Qirim. Investigating bias in facial analysis systems: A systematic review. *IEEE Access*, 8:130751–130761, 2020.

[53] S. R. Khanal, P. Sharma, H. Fernandes, J. Barroso, and V. M. d. J. Filipe. Performance analysis and evaluation of cloud vision emotion APIs. *Khanal, Salik Ram and Sharma*, 2023.

[54] V. Koufi, F. Malamateniou, and G. Vassilacopoulos. Towards clinical and operational efficiency through healthcare process analytics. *Int. J. Big Data Anal. Healthc.*, 1(1): 1–17, Jan. 2016.

[55] A. Koul, S. Ganju, and M. Kasam. *Practical Deep Learning for Cloud, Mobile, and Edge: Real-World AI & Computer-Vision Projects Using Python, Keras & TensorFlow*. O'Reilly Media, Inc., 2019. ISBN 9781492034858. URL `https://www.oreilly.com/library/view/practical-deep-learning/9781492034858/`.

[56] O. A. Kozlova and L. P. Kozlova. Image recognition systems in augmented reality systems. In *2022 Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus)*. IEEE, Jan. 2022.

[57] A. Kubany, S. Ben Ishay, R. S. Ohayon, A. Shmilovici, L. Rokach, and T. Doitshman. Comparison of state-of-the-art deep learning APIs for image multi-label classification using semantic metrics. *Expert Syst. Appl.*, 161(113656):113656, Dec. 2020.

[58] S. Kuutti, S. Fallah, R. Bowden, and P. Barber. Deep learning for autonomous vehicle control: Algorithms, state-of-the-art, and future prospects. *Synth. Lect. Adv. Automot. Technol.*, 3(4):1–80, Aug. 2019.

[59] K. N. Lam, N. N. Le, and J. Kalita. Building a chatbot on a closed domain using RASA. In *Proceedings of the 4th International Conference on Natural Language Processing and Information Retrieval*, New York, NY, USA, Dec. 2020. ACM.

[60] T. S. Madhulatha. An overview on clustering methods. *IOSR J. Eng.*, 02(04): 719–725, Apr. 2012.

[61] M. Matsangidou and J. Otterbacher. What is beautiful continues to be good: People images and algorithmic inferences on physical attractiveness. In *IFIP Conference on Human-Computer Interaction*, pages 243–264. Springer International Publishing, Cham, 2019.

[62] Microsoft. Microsoft computer vision api pricing, alternatives and more 2024 - saasworthy. `https://www.saasworthy.com/product/microsoft-computer-vision-api/pricing`, 2024. Accessed: 2024-02-10.

[63] Microsoft. Azure computer vision faq - microsoft. `https://learn.microsoft.com/en-us/azure/cognitive-services/computer-vision/faq`, 2024. Accessed: 2024-02-10.

[64] Microsoft. Azure computer vision overview - microsoft. `https://learn.microsoft.com/en-us/azure/ai-services/computer-vision`, 2024. Accessed: 2024-02-10.

[65] V. Muthukumar, T. Pedapati, N. Ratha, P. Sattigeri, C.-W. Wu, B. Kingsbury, A. Kumar, S. Thomas, A. Mojsilovic, and K. R. Varshney. Understanding unequal gender classification accuracy from face images. *Understanding unequal gender*, 2018.

[66] Oreil. Zebra image. Flickr, 2024. URL `https://www.flickr.com/photos/glenirah/2781264490`. Online; accessed 11-February-2024.

[67] F. Patrick. Computer vision documentation - quickstarts, tutorials, api reference - azure cognitive services. *Computer Vision documentation*, 2023.

[68] P. Prakrankamanant and E. Chuangsuwanich. Tokenization-based data augmentation for text classification. In *2022 19th International Joint Conference on Computer Science and Software Engineering (JCSSE)*. IEEE, June 2022.

[69] Priya and K. Shivarama Rao. Design and development of InfoRef using dialogflow: A 24/7 online reference service platform. *srels*, pages 185–190, July 2023.

[70] K. Ravneet and V. S. Dharam. Punjabi text recognition system for portable devices: A comparative performance analysis of cloud vision API with tesseract". en, pa. *Journal of Computer Science and Engineering*, pages 104–111, 2021.

[71] A. Sankar, A. Suresh, P. Varun Babu, A. Baskar, and S. K. Vasudevan. An in-depth analysis of applications of object recognition. *Res. J. Appl. Sci. Eng. Technol.*, 10 (1):1–14, May 2015.

[72] C. Senpeng, D. Jingwei, and D. Kaiying. On optimisation of web crawler system on scrapy framework. *Int. J. Wirel. Mob. Comput.*, 18(4):332, 2020.

[73] M. Serra-Burriel and C. Ames. Machine learning-based clustering analysis: Foundational concepts, methods, and applications. In *Acta Neurochirurgica Supplement*, Acta neurochirurgica. Supplement, pages 91–100. Springer International Publishing, Cham, 2022.

[74] V. Sharma. Object detection and recognition using amazon rekognition with boto3. In *2022 6th International Conference on Trends in Electronics and Informatics (ICOEI)*. IEEE, Apr. 2022.

[75] C. Sinan, S. Sachio, and N. Masahide. Toward flexible and efficient home context sensing: Capability evaluation and verification of Image-Based cognitive APIs". en. In *Number: 5 Publisher*, volume 20. Multidisciplinary Digital Publishing Institute, 2020.

[76] J. Snow. Amazon's face recognition falsely matched 28 members of congress with mugshots. `https://www.aclu.org/news/privacy-technology/amazons-face-recognition-falsely-matched-28`, July 2018. Accessed: 2023-9-12.

[77] K. Thammarak, P. Kongkla, Y. Sirisathitkul, and S. Intakosum. Comparative analysis of tesseract and google cloud vision for thai vehicle registration certificate. *Int. J. Electr. Comput. Eng. (IJECE)*, 12(2):1849, Apr. 2022.

[78] S. Thivaharan., G. Srivatsun., and S. Sarathambekai. A survey on python libraries used for social media content scraping. In *2020 International Conference on Smart Electronics and Communication (ICOSEC)*. IEEE, Sept. 2020.

[79] A. Tiwari. Data extraction from images through OCR. *Int. J. Res. Appl. Sci. Eng. Technol.*, 9(VIII):435–437, Aug. 2021.

[80] G. Uddin and F. Khomh. Automatic mining of opinions expressed about APIs in stack overflow. *IEEE Trans. Softw. Eng.*, 47(3):522–559, Mar. 2021.

[81] P. van den Besselaar and U. Sandström. What is the required level of data cleaning? a research evaluation case. *J. Sci. Res.*, 5(1):07–12, Apr. 2016.

[82] J. Van den Broeck, S. Argeseanu Cunningham, R. Eeckels, and K. Herbst. Data cleaning: Detecting, diagnosing, and editing data abnormalities. *PLoS Med.*, 2(10): e267, Sept. 2005.

[83] U. Yadav, S. Verma, D. K. Xaxa, and C. Mahobiya. A deep learning based character recognition system from multimedia document. In *2017 Innovations in Power and Advanced Computing Technologies (i-PACT)*. IEEE, Apr. 2017.

[84] J. Zeng, B. Flanagan, T. Sakai, and S. Hirokawa. Extraction of relevant components using shallow structure of HTML documents. In *2012 9th International Conference on Fuzzy Systems and Knowledge Discovery*. IEEE, May 2012.

[85] N. Zhang, Y. Zou, X. Xia, Q. Huang, D. Lo, and S. Li. Web APIs: Features, issues, and expectations – a large-scale empirical study of web APIs from two publicly accessible registries using stack overflow and a user survey. *IEEE Trans. Softw. Eng.*, 49(2):498–528, Feb. 2023.

A. Appendix

A.1. Decision Support Tool Implementation

```
1
2 def preprocess(text):
3     return [word for word in word_tokenize(text) if word.isalpha()]
```

Listing A.1: Cleans up the text by selecting only alphabetical words, useful for standardizing text for analysis.

```
1
2 def word_to_sentence_similarity(word, sentence):
3     try:
4         # Preprocess the word and sentence
5         word = preprocess(word)[0]
6         sentence = preprocess(sentence)
7         # Get the vector for the word
8         word_vector = model[word]  # Assuming word is a single word
9         # Get the vectors for the words in the sentence
10        sentence_vectors = np.array([model[w] for w in sentence if w in model.
      words])
11        # Compute the dot product and norms
12        dot_product = np.dot(word_vector, sentence_vectors.T)
13        word_vector_norm = np.linalg.norm(word_vector)
14        sentence_matrix_norm = np.linalg.norm(sentence_vectors, axis=1)
15        # Compute the average vector for the sentence
16        similarities = dot_product / (word_vector_norm * sentence_matrix_norm)
17        is_similar = max(similarities) >= threshold
18        return is_similar, max(similarities)
19    except Exception as e:
20        warnings.warn(f"Error in word_to_sentence_similarity: {e}")
21        return False, 0.0]
```

Listing A.2: Measures how closely a word is related to a given sentence, which can help in understanding the relevance of web content to specific keywords.

```
1
2 def is_valid(curr_url, base_domain):
3     parsed = urlparse(curr_url)
4     return bool(parsed.netloc) and parsed.netloc == base_domain]
```

Listing A.3: Verifies if a given URL belongs to the base domain of interest, ensuring that the scraping stays within the relevant domain.

```
1
2 def is_download_url(curr_url):
3     try:
4         response = requests.head(curr_url, allow_redirects=True)
5         content_type = response.headers.get('Content-Type', '')
```

```
        content_disposition = response.headers.get('Content-Disposition', '')

        if 'attachment' in content_disposition:
            return True
        elif any(file_type in content_type for file_type in ['application/pdf', '
    application/zip', 'image/', 'video/', 'audio/']):
            return True
        return False
    except requests.RequestException as e:
        print(f"An error occurred: {e}")
        return False]
```

Listing A.4: Determines if a URL points to a downloadable file, such as a PDF or ZIP, which can be crucial for focusing on text content and avoiding unnecessary downloads.

```

def scrape_page(curr_url, base_domain, keywords):
    global level
    if curr_url in visited:
        return

    if is_download_url(curr_url):
        return
        # Send an HTTP request to the URL
    response = requests.get(curr_url)
    if response.status_code != 200:
        print(f"Failed to get URL. Status code: {response.status_code}")
        return
    try:
        visited.add(curr_url)
        print(f"Level {max_level - level} - {curr_url} : success")
        level = level - 1
        # Initialize a BeautifulSoup object and specify the parser
        soup = BeautifulSoup(response.text, 'html.parser')

        # Logic to scrape the documentation
        paragraphs = soup.find_all('p')
        for i, paragraph in enumerate(paragraphs):
            text = paragraph.text
            if any(keyword in text for keyword in programming_keywords) or "\\u"
    in text:
                continue
            for keyword in keywords:
                text = " ".join([t for t in text.replace('\t', ' ').replace('\n',
    ' ').split(' ') if len(t) > 0])
                is_similar, sim_score = word_to_sentence_similarity(keyword, text)
                if is_similar and text not in unique_paragraphs and len(text.split
    (' ')) > 4:
                    unique_paragraphs.add(text)
                    sim_score = float(sim_score) if isinstance(sim_score, np.
    float32) else sim_score
                    categorized_data[keyword.lower()][text] = sim_score

        # Find internal links in the page and scrape them
        anchors = soup.find_all('a')
        for anchor in anchors:
            href = anchor.get('href')
            if href:
                next_url = urljoin(curr_url, href)
                if is_valid(next_url, base_domain) and level > 0:
```

```
                    scrape_page(next_url, base_domain, keywords)
    except Exception as e:
        print(f"Error parsing the content from {curr_url}: {e}")]
```

Listing A.5: Automates the process of visiting a webpage, extracting useful text, and navigating through internal links. This is the core of a web scraping operation that collects data for analysis.

```

def save_to_file(sub_directory, root_directory='documentation'):
    # Create directory if not exists
    if not os.path.exists(root_directory):
        os.makedirs(root_directory)

    full_path = os.path.join(root_directory, sub_directory)
    if not os.path.exists(full_path):
        os.makedirs(full_path)

    for category, dict_data in categorized_data.items():
        sorted_data = sort_dict_by_values(dict_data)
        # Open a separate file for each category
        with open(f'{full_path}/{category}.json', 'w', encoding='utf-8') as f:
            json.dump(sorted_data, f)]
```

Listing A.6: Organizes and stores the scraped and processed information into files, which is essential for data management and later retrieval.

```

def read_json_file(filepath):
    if not os.path.exists(filepath):
        # The file does not exist
        return None
    with open(filepath, 'r') as file:
        try:
            data = json.load(file)
        except json.JSONDecodeError:
            # JSON is empty or not properly formatted
            return None
    return data]
```

Listing A.7: Reads and returns the content of a JSON file, allowing for easy access to structured data that's been previously saved.

```

def sort_dict_by_values(d):
    return dict(sorted(d.items(), key=lambda item: item[1], reverse=True))]
```

Listing A.8: Orders dictionary entries by their values, which is helpful for prioritizing data based on certain metrics like relevance scores.

```

def find_highest_value_in_json(json_data):
    # Check if the json_data is empty or not a dictionary
    if not json_data or not isinstance(json_data, dict):
        return None, None
    highest_key = None
    highest_value = None
    for key, value in json_data.items():
```

```
        if highest_value is None or value > highest_value:
            highest_value = value
            highest_key = key
    return highest_key, highest_value]
```

Listing A.9: Identifies the most hight value in a JSON dictionary, which could be critical for highlighting the hightest similarity score in a dictionary data.

```

def process_job(json_files):
    for json_file in json_files:
        print(f"\nHighest values in {json_file}:")
        for api in apis.keys():
            filepath = os.path.join(doc_directory, api, f"{json_file}.json")
            json_data = read_json_file(filepath)
            if json_data is not None:
                key, value = find_highest_value_in_json(json_data)
                if value is not None:
                    print(f"The highest value in {api} {json_file} is {value} (key
    : {key})")
                else:
                    print(f"No valid data found for {api} {json_file} or file does
     not exist.")
            else:
                print(f"No data found for {api} {json_file}.")]
```

Listing A.10: Iterates through JSON files and utilizes the aforementioned function to find and report the highest similarities, according to the catgory selected by user.

```

def execute_with(curr_url, save_directory):
    base_domain = urlparse(curr_url).netloc  # Extract the base domain to ensure
    we stay on the same site
    scrape_page(curr_url, base_domain, categorized_data.keys())
    save_to_file(save_directory)  # Save categorized data to separate files]
```

Listing A.11: Drives the web scraping process based on a starting URL and saves the categorized data, streamlining the collection of web content.

```

def validate_indices(input_indices, max_index):
    indices = []
    for i in input_indices.split(','):
        i = i.strip()
        if not i.isdigit() or not 0 <= int(i) <= max_index:
            print(f"Invalid i: {i}. Please enter numeric indices within the valid
    range.")
            return None
        indices.append(int(i))
    return indices]
```

Listing A.12: Ensures that user input falls within a valid range, an essential step in user interaction to prevent errors in data selection processes..

```

if __name__ == '__main__':
```

```
    programming_keywords = ['var ', 'let ', 'const ', ';', '->', '::', '#include', 'import ',
                            'from ', 'def ', 'public:', 'private:', 'protected:', 'void ', 'parameter', 'struct {']

    all_json_files = ["performance", "cost", "limit", "functionality"]
    doc_directory = 'documentation'
    apis = {
        "imagga": "https://imagga.com",
        "cloudGoogle": "https://cloud.google.com/vision/docs",
        "amazon": "https://docs.aws.amazon.com/rekognition/latest/dg/what-is.html",
        "azure": "https://learn.microsoft.com/en-us/azure/ai-services/computer-vision/",
        "clarifai": "https://docs.clarifai.com/"
    }
    highest_values = {}
    max_level = 1000
    threshold = 0.5

    reload_data = input("Do you want to reload the documentation ? (yes/no): ").strip().lower()
    if reload_data == 'yes' or reload_data == 'y':
        model_path = hf_hub_download(repo_id="facebook/fasttext-en-vectors", filename="model.bin")
        model = fasttext.load_model(model_path)
        for name, url in apis.items():
            categorized_data = {"performance": {}, "cost": {}, "limit": {}, "functionality": {}}
            visited = set()
            unique_paragraphs = set()
            level = max_level
            execute_with(url, name)

    # Present options to the user
    print("Select the JSON files you want to compare by providing indices separated by commas:")
    for index, name in enumerate(all_json_files):
        print(f"{index} {name}")
    user_input = input("Enter your choice (e.g., 0,2): ")
    selected_indices = validate_indices(user_input, len(all_json_files) - 1)
    # If the validation fails, exit the script
    if selected_indices is None:
        print("Exiting due to invalid input.")
        exit()
    selected_json_files = [all_json_files[index] for index in selected_indices]
    process_job(selected_json_files)
]
```

Listing A.13: The central function in the previous diagram orchestrates the core process flow of the tool 4.2. It coordinates various tasks such as data reloading, web scraping, text processing, and information categorization, all while ensuring the relevance and accuracy of the data collected.

```
{
  "Face Liveness Detection - Rekognition Face Liveness is a fully managed machine learning (ML) feature designed to help developers deter fraud during face-based identity verification. The feature helps you verify that a user is physically present in front of the camera and isn\u00e2\u0080\u0099t a bad actor spoofing the user's face. Using Rekognition Face Liveness can help you
```

```
    detect spoof attacks presented to a camera, such as printed photos, digital
    photos/videos, or 3D masks. It also helps detect spoof attacks that bypass a
    camera, such as pre-recorded or deepfake videos injected directly into the
    video capture subsystem.": 1.0,
  "You can add features that detect objects, text, unsafe content, analyze images/
    videos, and compare faces to your application using Rekognition's APIs. With
    Amazon Rekognition's face recognition APIs, you can detect, analyze, and
    compare faces for a wide variety of use cases, including user verification,
    cataloging, people counting, and public safety.": 0.9410092234611511,
  "Face liveness - Detect if a live user is present during face verification.":
    0.9410092234611511,
  "Face-Based User Identity Verification - Confirm user identities by comparing
    faces in images to reference face images. Useful for identity verification in
    applications.": 0.9410092234611511,
  "Facial Search - With Rekognition, you can search images, stored videos, and
    streaming videos for faces that match those stored in a container known as a
    face collection. A face collection is an index of faces that you own and
    manage. Searching for people based on their faces requires that you Index the
    faces and then Search for the faces.": 0.9410092234611511,
  "Facial Analysis - Detect, analyze, and compare faces, along with facial
    attributes, such as gender, age, and emotions. Use cases may include user
    verification, cataloging, people counting, and public safety.":
    0.7269706130027771,
  "Facial Analysis - Detect, analyze, and compare faces in streaming or stored
    videos.": 0.7269706130027771,
  "Deep learning-based image and video analysis - Analyzes images and videos using
     deep learning for high accuracy. Amazon Rekognition' can detect labels,
    objects, scenes, faces, celebrities. Filter the results to include/exclude
    specific labels.": 0.7269706130027771,
  "Text Detection - Detect and extract text in images for visual search or
    extracting metadata. This works on different fonts and styles. Detects
    orientation to handle text on signs and banners.": 0.47309812903404236,
  "Unsafe Content Detection - Detect and filter explicit, inappropriate, and
    violent content in images and videos. Uses labels for granular filtering based
     on business needs. The Content Moderation API also returns a hierarchical
    list of any detected labels (objects and concepts), along with confidence
    scores. These objects/labels indicate specific categories of unsafe content,
    which enables granular filtering and management of large volumes of user-
    generated content (UGC). You can customize the output of the Content
    Moderation API with adapters, which enhance performance for images like those
    you provide as training data.": 0.4382784068584442,
  "Amazon Rekognition is a cloud-based image and video analysis service that makes
     it easy to add advanced computer vision capabilities to your applications.
    The service is powered by proven deep learning technology and it requires no
    machine learning expertise to use. Amazon Rekognition includes a simple, easy-
    to-use API that can quickly analyze any image or video file that\u00e2\u0080\
    u0099s stored in Amazon S3.": 0.43780672550201416,
  "To use the Amazon Web Services Documentation, Javascript must be enabled.
    Please refer to your browser's Help pages for instructions.":
    0.399850994348526,
  "Integrating powerful image and video analysis into your app - Add accurate
    image and video analysis to apps without expertise. The Amazon Rekognition API
     enables analysis via deep learning without requiring any machine learning
    knowledge. You can quickly build computer vision into web, mobile, and device
    apps.": 0.3922666311264038,
  "How Amazon Rekognition works \u00e2\u0080\u0093 This section introduces various
     Amazon Rekognition components that you work with to create an end-to-end
    experience.": 0.3778708577156067,
  "Detection of Personal Protective Equipment - Detect personal protective
    equipment in images to monitor safety compliance across various industries.
    You can automatically flag unsafe conditions by detecting improper equipment
    and receive alerts about these conditions, which can improve compliance and
```

```
    training.": 0.3729243874549866,
  "Custom Labels - Identify custom objects, concepts and scenes specific to
    business use cases, such as logo detection. You can train custom classifiers
    to handle niche or proprietary objects, which improves accuracy on key objects
     versus general classifiers. For more information, see What is Amazon
    Rekognition Custom Labels? in the Amazon Rekognition Custom Labels Developer
    Guide.": 0.3729243874549866,
  "Working with stored video analysis \u00e2\u0080\u0093 This section provides
    information about using Amazon Rekognition with videos stored in an Amazon S3
    bucket.": 0.36468783020973206,
  "Working with streaming video events \u00e2\u0080\u0093 This section provides
    information about using Amazon Rekognition with streaming videos.":
    0.36468783020973206,
  "Celebrity Recognition - Recognize celebrities in your images and videos across
    categories, such as politicians, athletes, actors, and musicians. You can
    identify celebrity appearances without needing to provide names.":
    0.35668182373046875,
  "Simple customization - Customize accuracy for your use case with adapters.
    Provide sample images to train adapters. Improves object and label detection
    for given domains. Easy way to tailor analysis without ML expertise.":
    0.3536853492259979,
  "Custom Labels - Build custom classifiers to detect objects specific to your use
     case, such as logos, products, characters.": 0.3457011580467224,
  "People pathing - Track identified people as they move across video frames.":
    0.3455717861652374,
}
```


Listing A.14: A JSON code snippet showing how data is saved.


```
{
    "cost": [
        "price", "cost", "expense", "fee", "charge", "pricing model",
        "subscription", "payment", "pay", "payment", "billing",
        "quote", "rate", "tariff", "budget", "financial",
        "affordable", "premium", "calculation", "fees", "economical",
        "monthly fee", "annual fee", "per image cost", "freemium",
        "trial period", "discount", "enterprise pricing",
        "custom pricing", "pay as you go", "hidden fees"
    ],
    "limit": [
        "limit", "restriction", "cap", "boundary", "quota",
        "maximum", "threshold", "upper limit", "limit per month",
        "limit per day", "rate limit", "API limit",
        "usage limit", "service limit", "capacity", "constraint",
        "policy", "usage restriction", "call limit", "request limit",
        "data cap", "bandwidth limit", "free tier limit",
        "concurrent request limit"
    ],
    "performance": [
        "performance", "speed", "efficiency", "throughput",
        "response time", "latency", "accuracy",
        "precision", "recall", "F1 score", "benchmark",
        "optimization", "scalability", "uptime", "downtime",
        "processing time", "resource usage", "load handling",
        "execution time", "inferencing speed", "real time processing",
        "batch processing speed", "failure rate", "detection rate",
        "recognition rate", "error rate", "performance metrics",
        "benchmark tests"
    ],
    "functionality": [
        "feature", "functionality", "characteristic", "attribute",
        "capability", "support", "integration",
```

```
        "recognition accuracy", "detection rate", "real time",
        "batch processing", "multi language support", "OCR",
        "object detection", "facial recognition", "labeling",
        "tagging", "metadata extraction", "custom models",
        "pre trained models", "analytics", "visualization",
        "API documentation", "model training", "data augmentation",
        "image segmentation", "text extraction", "landmark detection",
        "image classification", "image similarity", "image search",
        "content moderation", "face matching", "emotion detection",
        "pose estimation", "activity recognition",
        "contextual analysis"
    ]
    }
```

Listing A.15: A JSON code snippet showing the default category used in the tool.