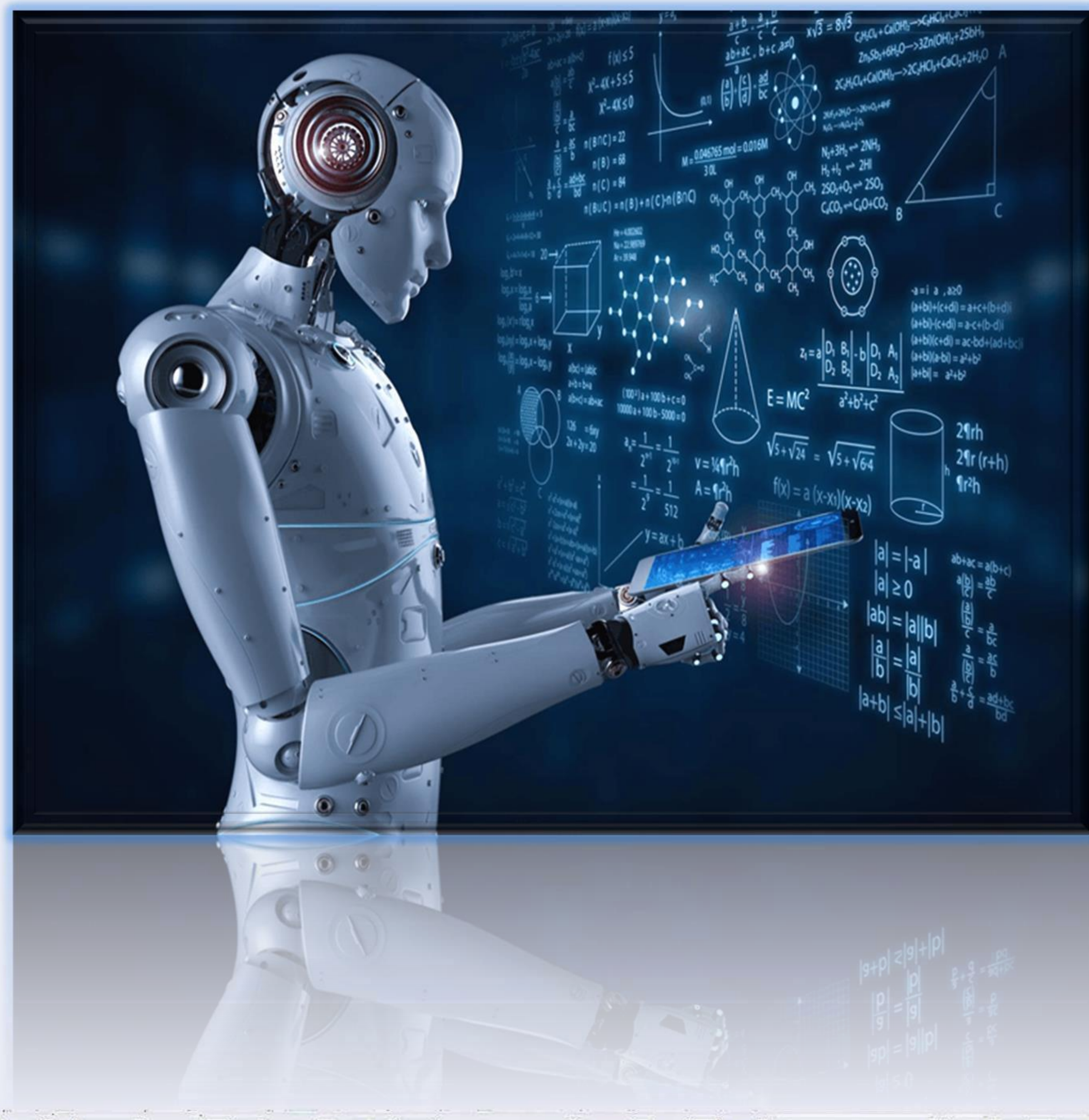
JCSM

Journal of Computer Sciences and Mathematics

ISSN 2955-246X Vol. 1 (2/2025)



JCSM

Journal of Computer Sciences and Mathematics

ISSN 2955-246X

Vol.1 (2/2025)

Publisher: Mother Teresa University in Skopje, Republic of North Macedonia

ISSN 2955-246X **Technical Editing/Layout:** Korab Ballanca

Editorial Office: Mother Teresa University

Frequency: Published two times per year

ISSN 2955-246X

Tax No. 4080016561272

Bank: Narodna Banka RM

Mother Teresa University in Skopje, Republic of North Macedonia
Mirce Acev 4, VII floor, Skopje, North Macedonia

Phone: +389 2 3161 004

E-mail: journals@unt.edu.mk
Web: www.journals.unt.edu.mk

The publication of the Journal is supported by:

Ministry of Culture of Republic of North Macedonia

Editorial Foreword

Editor-in-Chief

ACKNOWLEDGEMENT

Welcome to the new Issue of the Journal Of Computer Sciences and Mathematics JCSM. The topics covered by this Issue are related to the current trends of research, original research that uncovers sustainable development.

The JCSM Journal represents an international journal that effectively provides a forum for academics, professionals, graduate and undergraduate students, fellows and associates to share the latest developments and advances in knowledge and practice of Computer Sciences, Artificial Intelligence, Cybersecurity, Information Technology and Engineering, Technics and Technology implications in Businesses. Our interest in promoting high-quality research is clearly reflected in having an established peer reviewing process and a high-profile expert group of Associate Editors and Editorial Board Members.

Hopefully you find this Issue valuable and we definitely look forward to receiving your high-quality studies for the next issue of the Journal.

Prof. Dr. Bekim Fetaji
Editor

Journal of Computer Science and Mathematics

Table of Contents

The use of Artificial Intelligence for Creating Digital Content in the Learning Process

Maja Kukusheva Paneva†

Faculty of Electrical Engineering, Goce Delcev University, Stip, N. Macedonia, maja.kukuseva@ugd.edu.mk

Aleksandar Velinov

Faculty of Computer Science, Goce Delcev University, Stip, N. Macedonia, aleksandar.velinov@ugd.edu.mk

Aleksandra Nikolova

Faculty of Computer Science, Goce Delcev University, Stip, N. Macedonia, aleksandra.nikolova@ugd.edu.mk

Natasha Stojkovikj

Faculty of Computer Science, Goce Delcev University, Stip, N. Macedonia, natasa.stojkovik@ugd.edu.mk

Zoran Zdravev

Faculty of Computer Science, Goce Delcev University, Stip, N. Macedonia, zoran.zdravev@ugd.edu.mk

† Corresponding author.

ABSTRACT

In the paper the capabilities of artificial intelligence (AI) in generating digital content for use in the learning process with focus on the course Fundamentals of Electrical Engineering are explored. Using three modern AI tools- ChatGPT, Copilot and Gemini, power point presentations are created to analyze the different approaches each tool takes in generating educational materials. The goal is to evaluate the structure, most relevant content, useful examples and the most effective visual elements for the need of electrical engineering students.

Through the research, each AI tool is used to create presentations on various topics from the course, such as fundamental electrical laws, electrical circuits, network theorems and dc transients. The presentations are compared based on several criteria: (1) information structure- how clearly and logically the content is organized; (2) quality of the content- accuracy, detail and relevance of the material; (3) examples- whenever and how the tool generates examples to explain the concepts; and (4) visibility and shareability.

The paper analyzes the strengths and weaknesses of each tool, with a particular focus on their usefulness for educators and students. Also, a discusses how AI tools can assist in improving learning by automating content creation while highlighting potential limitations will be represented. The results of this research will be valuable to the academic community, offering insights into which AI tool provides the best educational materials.

KEYWORDS

Artificial Intelligence, ChatGPT, Gemini, Copilot, education, student.

1 Introduction

The rapid advancement of artificial intelligence (AI) is transforming various sectors, and education is no exception. In recent years, AI tools have increasingly been used to enhance the teaching and learning process by automating the creation of educational materials, [1], [2]. These technologies are being integrated into a range of platforms to support the professors in generating personalized, accurate and engaging content for students, [3]. The impact of artificial intelligence (AI) is particularly promising in creation of digital presentation.

This paper examines the use of AI tools for generating digital content for the course Fundamentals of Electrical Engineering. This course covers complex topics where clear and well-structured presentations are critical for student comprehension. To explore the effectiveness of AI, three AI-tools- ChatGPT, Copilot and Gemini, were used to create Power Point presentation on same various courses topics.

The purpose of this research is to analyze and compare the output presentation in terms of structure, content quality, included examples and visual elements. Additionally, the paper addresses the visibility and shareability of the AI- generated presentation, emphasizing how easily these materials can be distributed and accessed by students.

By evaluating the strengths and limitations of each AI tool, this research aims to provide insights into how AI- generated content can enhance the learning process in technical fields like electrical engineering. Ultimately, the findings will offer valuable guidance to professors and institutions seeking to incorporate AI into their curriculum development process.

2 Literature review

In recent years, the integration of Artificial Intelligence (AI) into education has been widely explored with studies emphasizing its potential to enhance learning and teaching process. AI tools have been particularly impactful in automating content creation, personalizing learning experiences and supporting teachers and professors by reducing their workload. AI has increasingly been employed to improve the accessibility and quality of education. Research highlights the use of AI- driven platforms to generate quizzes, create plans and develop interactive content. Studies such as in [1] discuss the potential of AI to revolutionize education by enabling adaptive learning environment by tailoring content to individual learner needs. Similarly, Holmes et al. [4] explore the implication of AI in education, discussing how it can transform teaching practices, improve learning outcomes and address the challenges associated with its integration.

Automating content generation is growing area of AI research, particularly in creating educational materials like presentations, summaries and tutorials. Tools like ChatGPT which leverage natural language processing have demonstrated capabilities in generating coherent and contextually relevant textual content. Research by OpenAI underlines ChatGPT's ability to assist in drafting educational resources with minimal user input. Copilot, a tool developed by GitHub, is primarily designed to assist programmers by suggesting code snippers. However, its use has been extended in generating technical content and explanations, as noted in studies on its role in educational programming environments, [5]. Meanwhile, Gemini is emerging AI tool that combines natural language processing with multimedia generation, making it a promising candidate for creating visually appealing educational content.

While existing literature highlights the general benefits of AI in education, limited studies focus on evaluating AI tools specifically for generating course materials in technical subjects like electrical engineering. This study addresses this gap by comparing ChatGPT, Gemini and Copilot in their ability to create Power Point presentations for the course Fundamental of electrical engineering.

3 Methodology

This study evaluates the capabilities of three modern AI tools- ChatGPT, Copilot and Gemini in generating digital content for educational purposes, specifically Power Point presentation for the course of Fundamental of electrical engineering. The methodology includes a systematic process of content creation, evaluation and comparison on predefined criteria.

Each AI tool was tasked with generating Power Point presentations on key topics from the course including fundamental of electrical laws, electrical circuits, network theorems and dc transients. The input guidelines provided for each tool were standardized to ensure consistency in output across all topics. The generated presentations were assessed based on following criteria:

(i) Information structure: how clearly and logically the content is organized.

(ii) Content quality: the accuracy, relevance and comprehensiveness of the material.

(iii) Examples: the inclusion of illustrative examples to explain concepts and their clarity.

(iv) Visual appeal and shareability: the effectiveness of visual elements and the ease of sharing the presentation.

4 Results

In this section a comparative analysis of the presentations generated by ChatGPT, Copilot and Gemini are represented highlighting their strengths and weaknesses across the specified evaluation criteria.

The first criterion is information structure of the Power Point presentation is evaluated based on the logical flow, clarity, use of headings and subheadings and overall ease of understanding for the students. ChatGPT demonstrated excellent organization and logical flow in the presentations. Each topic was divided into sections and subsections with clear headings and subheadings. For example, when creating a presentation on electrical circuits, ChatGPT structured the content into sequential slides covering definitions, key principles, examples and summary points making it suitable for theoretical and concept- heavy subjects. But, despite the strong organization, the presentations created with ChatGPT tend to be verbose often including excessive text that might overwhelm the students. On the other hand, Copilot concise and structure outputs focusing on technical aspects. For instance, when generating content on network theorems, Copilot divided the content into clear and well-defined sections for each theorem with brief explanation and examples. Its structure is excellent for highly technical content or numerical examples but has limited ability to independently structure non- programming content effectively. Gemini excels in balancing content organization and visual appeal. The tool organizes information into logical sections and pairs it with corresponding visuals, such as diagrams, charts, or graphics. For example, in presentations on DC transients, Gemini seamlessly structured the slides to introduce key concepts, followed by graphs illustrating voltage and current over time. While the structure is visually engaging, Gemini prioritizes design over logical flow. This may lead to slides that look appealing but require additional effort to understand the connections between sections.

The second criterion is content quality and was evaluated based on three key factors: accuracy, relevance and comprehensiveness of the material generated by these AI tools. ChatGPT demonstrated a high degree of accuracy in its textual explanations and the content was well- researched and the concepts were correctly defined. The content generated is highly relevant to the course topics and the examples provided facilitate the learning process of the students. The presentations generated by ChatGPT are comprehensive and to ease the studying process each topic is divided into smaller sections followed by examples. Copilot is excellent in technical accuracy especially when numerical problems or equations were involved and produces highly relevant content. However, its focus on concise technical outputs meant that contextual relevance was sometimes missing, making it less suited for introductory content. Copilot is less comprehensive compared to ChatGPT and it needs supplemental explanation on the topics. Content generated by Gemini was generally accurate but there were instances where the focus on visuals led to slight oversimplification or generalized explanations. However, Gemini sometimes simplified complex topics too much to fit its visually appealing design, which could leave gaps in understanding for advanced topics.

The third criterion is the use of examples that are critical components in evaluating the effectiveness of AI generated presentation. The examples help clarify complex concepts and enhance the learning process. ChatGPT, Copilot and Gemini generated the same examples. For example, on the topic Ohm's Law an example was generated for simple circuit with given voltage and resistance, calculating the current, as shown in Figure 1. But the generated examples from all three AI tools are generic and lack specificity to advanced electrical engineering topics, requiring further refinement to align with course objectives.

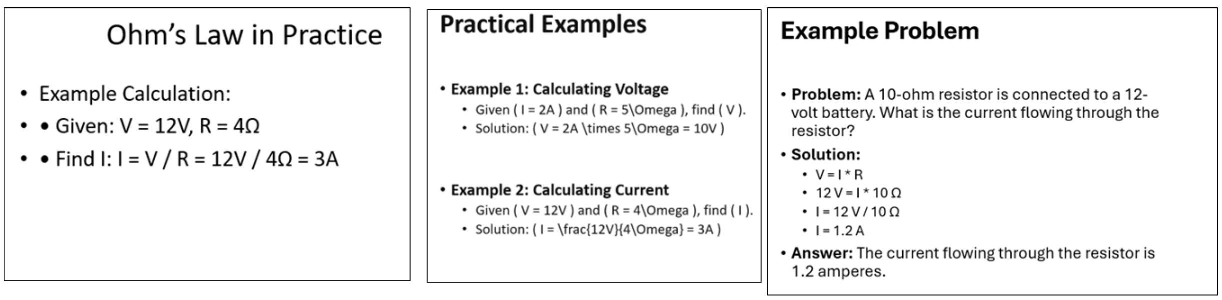


Figure 1: Generated example for Ohm's Law: (a) ChatGPT (b) Copilot (c) Gemini

The last criterion is visual appeal and shareability of the generated Power Point presentations. The visual appeal of educational materials plays a critical role in engaging students and ensuring distribution on the generated presentations. ChatGPT generated text-based content that can be customized for presentations. While it lacks built-in visual design capabilities, its output can be formatted into visually appealing slides using tools like PowerPoint or Canva. For example, ChatGPT provides structured bullet points and logical sequences that lend themselves to easy transformation into slide format. Outputs are flexible and can be adapted to various formats, including PowerPoint, PDF, and Word documents. This adaptability makes sharing across platforms straightforward. Copilot focuses on generating precise technical content, including equations and tables which are highly valuable for educational presentations. While it does not generate visuals, the structured output can be paired with graphs or diagrams to enhance the presentation. But the main weakness of Copilot is limited direct support for creating presentation-ready materials. Gemini excels in visual design, offering integrated support for diagrams, animations, and aesthetically pleasing layouts. For instance, when generating slides on DC transients, Gemini includes time-domain graphs and waveform illustrations that enhance understanding. Its ability to combine text and visuals into a cohesive design makes it particularly effective for topics that benefit from graphical representation, such as circuit diagrams or network theorems. The tool supports exporting presentations in multiple formats, including PowerPoint and PDF, ensuring compatibility with most platforms.

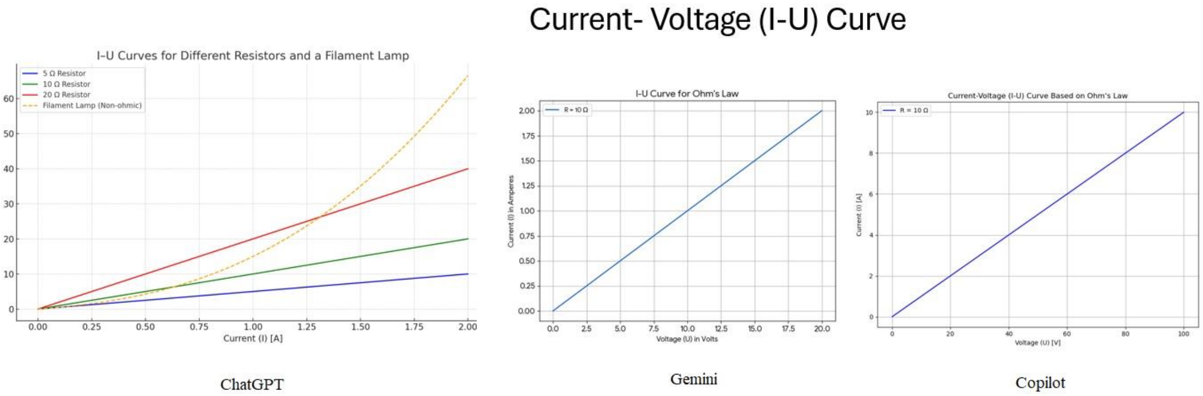


Figure 2: Generated current- voltage curve: (a) ChatGPT (b) Gemini (c) Copilot

Although ChatGPT and Gemini do not generate graphics and images within the presentation itself, these AI tools can subsequently generate images that could later be inserted into the presentation. For example, all three AI tools were given command to generate the current- voltage characteristic of Ohm's Law (Figure 2). The graphics generated by Copilot and Gemini accurately represent a linear relationship consistent with Ohm's Law. The graph generated by ChatGPT is the most comprehensive because it shows not only I-U curves for three different resistors but also includes lamp which correctly demonstrates a non- linear relationship.

5 Conclusion

This paper explores and evaluates the capabilities of three moder AI tools- ChatGPT, Copilot and Gemini, for generating digital presentation to support the learning process on the course Fundamental of electrical engineering. By comparing these tools across key criteria such as information structure, content quality, included examples, visual appeal and shareability, the study provides insights into their strengths, limitations and potential to enhance the educational materials for students.

ChatGPT performed best in information structure and content quality, offering logically organized, accurate and comprehensive material that is highly relevant to course objectives. However, its tendency toward verbosity and lack of built-I visuals may require manual adjustments to avoid overwhelming students. Copilot excelled in technical accuracy, especially for numerical problems and equations and produced clearly segmented content on each topic. However, it lacked depth and contextual completeness and requires additional visual presentation. Gemini stood out in visual appeal, effectively integrating diagrams and layout design to create engaging slides. However, its focus on aesthetics sometimes comes at the cost of depth and logical comprehensive for more advanced topics.

In summary, ChatGPT is most effective for generating structured, accurate, and comprehensive presentations, especially in theoretical domains. Copilot is ideal for technical precision, while Gemini offers superior visual presentation, best used when graphical illustration is a priority. Educators may consider blending outputs from these tools—using ChatGPT for core content, Copilot for technical depth, and Gemini for visual enhancement—to achieve the most effective learning materials.

REFERENCES

[1] Luckin, R., Holmes, W., Griffiths, M., and Forcier, L. B. (2016). Intelligence Unleashed: An Argument for AI in Education. *Pearson.*

[2] Kumar, S., and Jha, S. (2020). AI-based Approaches in Education: Leveraging Learning Analytics and Automated Content Generation. International Journal of Educational Management, 34 (7), 1354-1369.

[3] Baker, R. S., and Siemens, G. (2014). Educational Data Mining and Learning Analytics. Review of Research in Education, 39 (1), 57-82.

[4] Holmes, W., Bialik, M., and Fadel, C. (2019). *Artificial Intelligence in Education: Promises and Implications for Teaching and Learning*. Center for Curriculum Redesign

[5] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., ... & Amodei, D. (2021). Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33, 1877-1901.

Overview of Dew Computing

Stojan Kitanov

Faculty of Information Sciences, Mother Teresa University, Skopje, Republic of North Macedonia, stojan.kitanov@unt.edu.mk

Ivan Petrov

School of Computer Science and Information Technology, University American College Skopje, Skopje, Republic of North Macedonia, ivan.petrov@uacs.edu.mk

Kire Jakimoski

Military Academy “General Mihailo Apostolski” - Skopje, University Goce Delcev, Stip, Republic of North Macedonia, kire.jakimoski@ugd.edu.mk

Toni Janevski

Faculty of Electrical Engineering and Information Technologies, Ss Cyril and Methodius University, Skopje, Republic of North Macedonia, tonij@feit.ukim.edu.mk

ABSTRACT

The most significant amount of information processing all around us is done on the lowest possible computing level, outright connected to the physical environment and mostly directly controlling our human immediate surroundings. These "invisible", "embedded" information processing devices can be found in everything from our life and work. These devices, which are neither at the cloud/fog edge, nor even at the mobile edge, but rather at the physical edge of computing are the basis of the Dew Computing Paradigm, which can be considered a further extension of Cloud Computing. This paper underlines the basic ideas and concepts of dew computing, the dew computing architecture, as well as, its research trends and directions.

KEYWORDS

Cloud Computing, Distributed Computing, Edge, Computing, Fog Computing, IoT, Mobile Cloud Computing, Mobile Edge Computing

1 Introduction

The fundamental idea of cloud computing is based on a very fundamental principal of reusability of IT capabilities [1]. This computing paradigm provides dynamically scalable infrastructure for application, data, and file storage by connecting a vast pool of systems in either private or public networks. Networks, servers, data warehouses, apps, and services are examples of shared cloud resources that may be quickly supplied and managed with little involvement from service providers. These resources are available to cloud computing customers for the development, hosting, and flexible operation of services and applications at any time, on any device, and anywhere in the cloud. This technology has greatly lowered the cost of computation, application hosting, content storage, and distribution [2].

At first, cloud computing was seen to be a promising way to address scalability and flexibility problems. For instance, cloud computing considers making it easier for businesses to manage their computer resources when selecting service levels based on capabilities. Nevertheless, this also creates a new issue: the user cannot control any resources that are situated far from his home. Additionally, if a user loses an Internet connection, he will not be able to access his own data.

Therefore, cloud computing technology cannot advance on its own. Together with the rapid proliferation of mobile devices and the Internet of Things (IoT), the advantages of cloud computing technology are further extended and facilitated by promoting wider technological advancement, with which emerged a new research area, known as **dew computing** [3], and it defined as:

“Dew computing is an on-premises computer software-hardware organization paradigm in the cloud computing environment where the on-premises computer provides functionality that is independent of cloud services and is also collaborative with cloud services. The goal of dew computing is to fully realize the potentials of on-premises computers and cloud services. [3]”

Dew computing is another of the distributed computing paradigms which is considered as an extension of the cloud computing paradigm. By using local resources to meet processing demands, Dew Computing, an extension of edge computing, reduces reliance on centralized infrastructure. All of the data will be kept on the user's device's local storage, and synchronization will be done to update the data on cloud-based apps whenever an internet connection is available. Dew computing can be thought of as the implementation of distributed computing systems at the lowest level. The system's proximity to the user is referred to as the lowest layer. Using the internet for data synchronization and enabling consumer devices as computing resources are top priorities for Dew Computing. Users of dew computing can utilize system capabilities without constantly being online. Dew computing makes use of local device resources to improve cloud services. For quicker, more effective offline apps, the workload is balanced between the cloud and the edge.

Despite challenges with interoperability and resource management, dew computing offers immense promise to satisfy the needs of persons with disabilities in a range of industries, including healthcare, smart cities, and the Internet of Things, revolutionizing the way computer resources are used. Dew computing, which puts resources closer to end users, functions independently of constant internet access, in contrast to typical cloud and edge computing [4], [5].

This paper outlines the basic ideas and concepts of dew computing, the dew computing architecture, as well as, its research trends and directions. It is organized as follows. Section 2 provides the evolution to dew computing. Section 3 provides a scalable hierarchical structure of dew computing. Section 4 provides the components of a dew computing architecture. Section 5 explains the possible application categories of dew computing. Finally section 6 concludes the paper and provides future research directions.

2 Evolution to a Dew Computing

The evolution timeline to a Dew Computing is described by Loncar in [6] and it is illustrated in Figure 1. It resulted due to the modern computing paradigms such as cluster computing, grid computing, cloud computing and fog computing, that are used in handling growing volume of data, providing computing and storage resources for such data-intensive applications. Each computing paradigm is described in [6].

By observing the Figure 1, it can be noticed that dew computing represents a "mini" version of a giant cloud computing architecture. The term "dew" can be analogized that the data stored in this system is "a dew" from the amount of data stored in the cloud computing system [7].

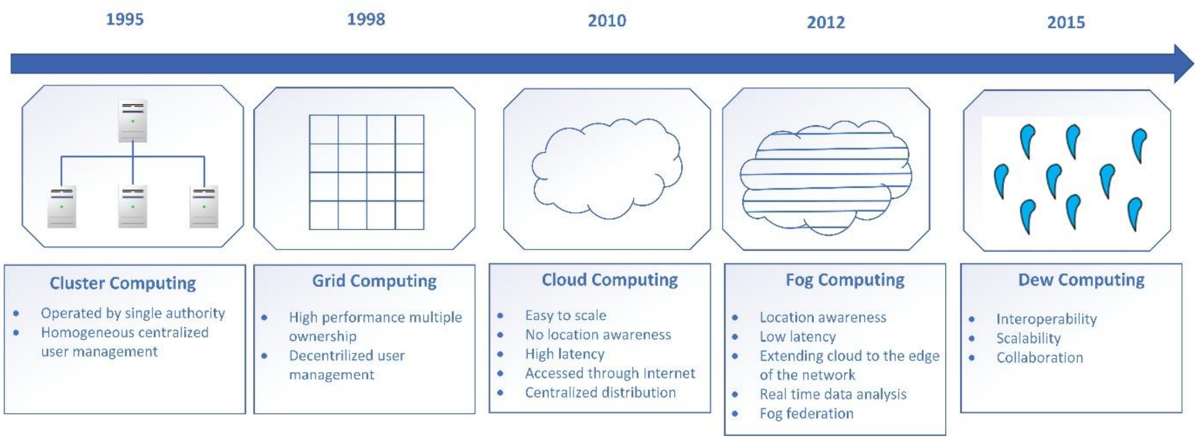


Figure 1: Dew Computing Evolution Timeline [6]

3 Scalable Hierarchical Structure of Dew Computing

The exponential expansion of data (Big Data) and processing (application) demands that we are currently seeing necessitates the scalability of resources on several levels. As a result, a hierarchical structure, consisting of three layers (Cloud, Fog and Dew Computing) was proposed in [8]. These hierarchical layers facilitate the rapidly developing complex distributed computer systems and meet the following requirements:

- **Performance:** optimized for fast responses, processing and low latency;
- **Availability**: requires redundancy, rapid recovery in the case of system failures, and graceful degradation when problems occur;
- **Reliability**: system needs to be reliable in data and function;
- **Manageability**: scalable system that is easy to operate;
- **Cost**: includes hardware and software costs but it is also important to consider other facets needed to deploy and maintain a scalable computing system.

This hierarchy of cloud, fog and dew computing layers, together with the structural, resource and application aspects is illustrated in Figure 3.

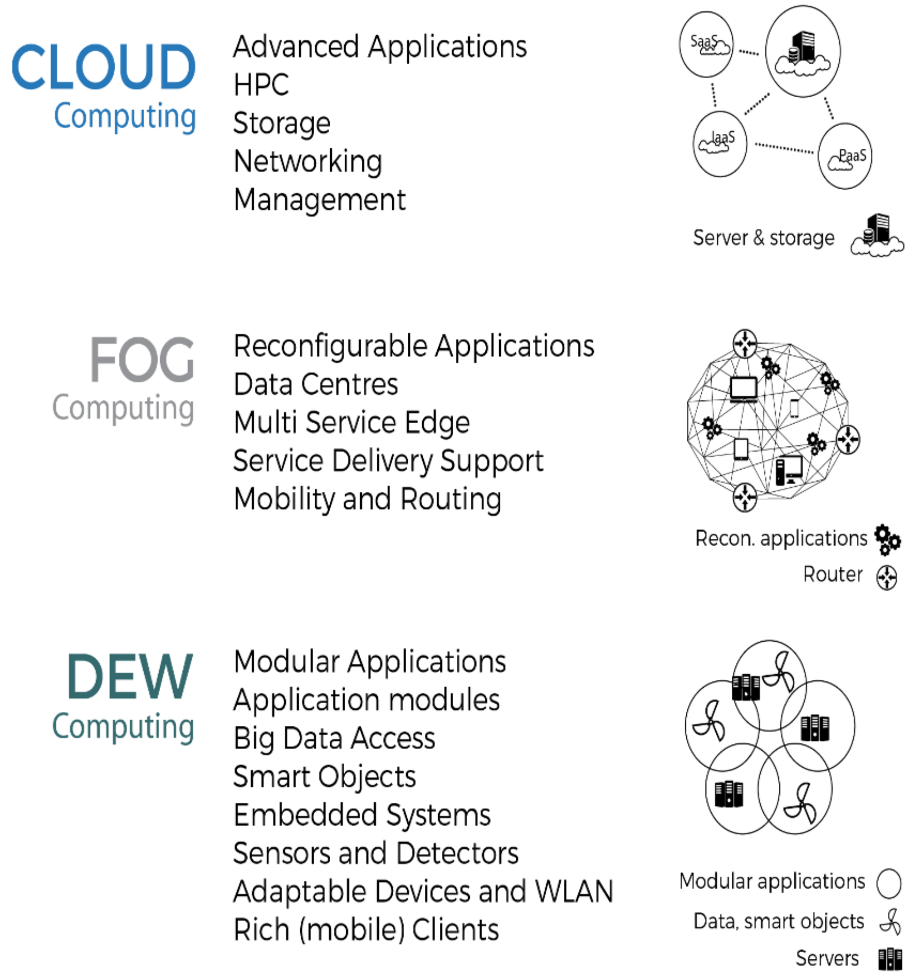


Figure 2: Hierarchy of Scalable Dew Computing [8].

4 Components in a Dew Computing Architecture

Dew computing architecture is illustrated on Figure 1, and contains a Dew Virtual Machine (DVM), which is the environment (computer or PC) where the dew computing system is running. DVM contains the following components [9]:

- **Dew Server (DS)**: it represents a small replica of a server in the cloud attached to a local computer. This component interacts periodically with cloud servers in the cloud and synchronizes with information on servers in the cloud.
- **Dew Analytic Server (DAS)**: is a local version of the analytics web server, which analyzes the data generated when system users use a dew server, or visit a dew server. The advantage of DAS is that user data is preprocessed before being sent to the server in the cloud.
- **Artificial Intelligence of Dew (AID)**: is a component which collects the results from the DAS, and guides operations on the dew server, and performs operational customization on the DS.

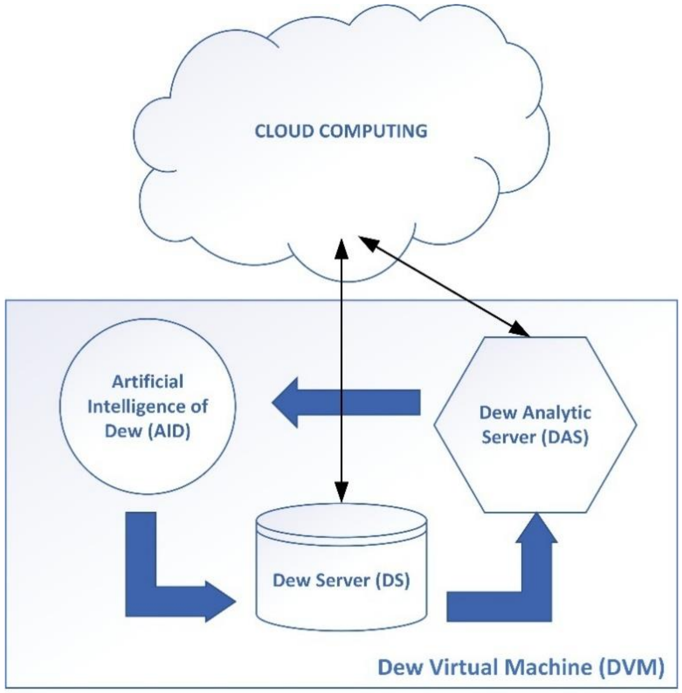


Figure 1: Components of a Dew Computing Architecture [9].

5 Application Categories of Dew Computing

Dew computing can be applied in different areas [3]. A summary of some possible dew computing categories is provided in Table 1. Each category is briefly described.

Web in Dew (WiD) is a dew computing category where an on-premises computer contains a duplicated fraction of World Wide Web, or a modified copy of that fraction [3]. A Web server should be launched and the relevant webpages should be made on the on-premises computer in order to accomplish this. WiD has great potentials in many areas, such as Internet of Things (IoT) applications [10].

Storage in Dew (STiD) is a dew computing category where the storage of an on-premises computer is duplicated in a cloud service and the on-premises computer storage is automatically synchronized with its cloud copy [3]. A typical example of an STiD service is Dropbox.

Database in Dew (DiD) is a dew computing category where a database on a cloud service and a database on an on-premises computer serve as backup to each other [3]. Users can choose which database is a replicated backup and which is the primary database. Additionally, there are several methods for doing the backup, including periodic log backup, real-time log backup, and real-time database replication. DBiD will make DBA's job much easier and the databases much safer.

Software in Dew (SiD) is a dew computing category where a user's ownership to a piece of software is not only reflected by the software's existence on the user's on-premises computer, but also reflected by the ownership and settings information recorded in a cloud service [3]. Additionally, SiD should ensure that the user can download the software again if needed. Examples of SiD are Apple's App Store and Google Play Store.

Platform in Dew (PiD) is a dew computing category where (1) a suite of software supporting the development and operation for specific purposes is installed and running on an on-premises computer; (2) the settings and application data of this software suite are dynamically synchronized with a cloud service [3]. An existing application which fulfills PiD requirements is GitHub.

Infrastructure as Dew (IaD) is a dew computing category where an on-premises computer is dynamically supported by cloud services. IaD can be used in the following two forms: (1) An on-premises computer can have an exact duplicate virtual machine instance in the cloud, and the virtual machine instance is always kept in the same state as the on-premises computer; (2) An on-premises computer can have all its settings/data saved in cloud services. Such settings/data not only include system settings/data, but also include each application's settings/data. If IaD is fully implemented, data and devices can be completely separated. If a laptop or a cell phone is lost or damaged, the user only needs to buy a new device and all the settings/data will be completely recovered in the new device.

Data in Dew (DiD) is a dew computing category where all its applications satisfy the independence and collaboration requirements, but they do not belong to any of the above categories [3].

Table 1: Summary of Dew Computing Categories and Applications [3].

Category	Resource in Dew	Key Function	Existing Applications
Web in Dew (WiD)	Web fraction	Access Web fraction without Internet connection	
Storage in Dew (STiD)	Storage	Storage in a dew has a cloud copy	Dropbox (2007)
Database in Dew (DBiD)	Database	Local database has a cloud backup	
Software in Dew (SiD)	Software	Software ownership and settings have a cloud copy	Apple App Store (2008), Google Play Store (2008)
Platform in Dew (PiD)	Platform suite	SDK and projects have a cloud copy	GitHub (2008)
Infrastructure as a Dew (IaD)	Whole computer	On-premises computer settings and data have a cloud copy	
Data in Dew (DiD)	Data in forms other than above	Dew computing applications not in above categories	Novell Groupwise (2005)

CONCLUSION

Dew computing is based on the vertically distributed computing classification's microservice concept. Distributed systems make it possible to offer a web browsing experience without an Internet connection. It utilizes local resources to improve cloud services and it enables efficient IoT operations during network disruptions. Information processing (raw data and metadata describing those data), high productivity of user requests (programmability/reconfigurability), and high equipment efficiency (complications of daily human information environment) are the three main focuses of Dew Computing.

Our Future research will focus on energy efficiency, scalability, and security in a dew computing environment.

REFERENCES

[1] Zhang, S., Zhang, S., Chen, X., and Huo, X. (2010). "Cloud Computing Research and Development Trend." Proceedings of the Second International Conference on Future Networks, ICFN'10, Sanya, China: 93-97. DOI: https://doi.org/10.1109/ICFN.2010.58.

[2] Nasir, V., and M. Bayramusta, M. (2016). "A fad or future of IT?: A comprehensive literature review on the cloud computing research," International Journal of Information Management, 36(4): 635-644.

[3] Wang, Y. (2016). "Definition and Categorization of Dew Computing," Open Journal of Cloud Computing, 3(1):1-7. DOI: urn:nbn:de:101:1-201705194546.

[4] Tyagi, A. K. (2024). ''Dew computing: State of the art, opportunities, and research challenges,'' in Machine Learning Algorithms Using Scikit and TensorFlow Environments: 332–345.

[5] Gusev, M. (2022). ''Scalable dew computing,'' Applied Sciences, 12(19): 9510. DOI: https://doi.org/10.3390/app12199510.

[6] Loncar, P. (2018). "Data-Intensive Computing Paradigms for Big Data," Proceedings of the 29th DAAAM International Symposium: 1010-1018, B. Katalinic (Ed.), Published by DAAAM International, ISBN 978-3-902734-20-4, ISSN 1726-9679, Vienna, Austria. DOI: http://dx.doi.org/10.2507/29th.daaam.proceedings.144.

[7] Utomo P., and Falahah, "Dew Computing: Concept and Its Implementation Strategy," 2020 Fifth International Conference on Informatics and Computing (ICIC), Gorontalo, Indonesia, 2020, pp. 1-6, DOI: https://doi.org/10.1109/ICIC50835.2020.9288581.

[8] Skala K., Davidovic D., Afgan, E. Sovic I., and Zorislav Sojat. (2015). "Scalable Distributed Computing Hierarchy: Cloud, Fog and Dew Computing," Open Journal of Cloud Computing (OJCC), 2(1): 16-24. DOI: 10.19210/1002.2.1.16, URN: urn:nbn:de:101:1-201705194519, GNL-LP: 1132360749.

[9] W. Yingwei, W. et.al. (2019). "Dew computing and transition of internet computing paradigms", ZTE Communications 15(4): 30-37.

[10] Palattella M., Dohler M., Grieco A., Rizzo G., Torsner J., Engel T. and Ladid L. (2016). "Internet of Things in the 5G Era: Enablers, Architecture and Business Models," IEEE Journal on Selected Areas in Communications, 34(3):1-17. DOI: https://doi.org/10.1109/JSAC.2016.2525418.

Computer Vision for Real-time Drowsiness Detection System. Case Study: Image Analysis

Mirlinda Ebibi
Faculty of Informatics, Mother Teresa University, Skopje, North Macedonia, mirlinda.ebibi@unt.edu.mk

Artan Ebibi
Faculty of Computer Science & Engineering, Skopje, North Macedonia, a.ebibi123@gmail.com

Abstract: In this paper, the realm of computer vision for real-time drowsiness detection in vehicles is explored, emphasizing the critical role of image capture techniques in ensuring driver safety. The primary objective of this research is to develop and refine algorithms that can accurately detect early signs of driver drowsiness by analyzing visual data captured from in-vehicle cameras. Through a comprehensive examination of existing methodologies and emerging technologies, this study seeks to improve both the precision and response time of drowsiness detection systems.

We will address key challenges in this domain, such as variable lighting conditions, head pose estimation, and facial landmark recognition, which are known to impact image quality and detection accuracy. By presenting innovative approaches to optimize image capture processes, including techniques for handling occlusions, low-light environments, and minimizing processing latency, this paper will highlight potential breakthroughs that can be applied to real-world vehicle environments.

Keywords: computer vision, drowsiness detection, image capture, optimization, real-time, vehicle safety.

I. Introduction

Drowsy driving continues to be a major factor in road accidents globally, posing a serious threat to public safety. The National Highway Traffic Safety Administration reports that drowsy driving leads to numerous deaths and injuries each year, underscoring the pressing need for effective preventive measures. While traditional methods like driver education and rest recommendations have been used to address this issue, they often fall short in delivering immediate interventions to reduce the risks associated with driver fatigue. Recent advancements in computer vision technology have introduced new opportunities for enhancing driver safety through automated drowsiness detection systems. By developing machine learning algorithms that analyze facial expressions, eye movements and other physiological indicators, these systems can identify signs of drowsiness in real time and issue timely alerts to drivers, potentially averting accidents proactively.

Despite the promise of computer vision-based solutions, there remains a gap in research regarding the optimization of image capture techniques for drowsiness detection in vehicles. The effectiveness of these systems relies on the quality and reliability of the captured images, which can be caused by factors like lighting conditions, camera positioning, and vehicle vibrations.This paper aims to address this gap by exploring computer vision techniques for optimizing image capture in real-time drowsiness detection systems.

We will address key challenges in this domain, such as variable lighting conditions, head pose estimation, and facial landmark recognition, which are known to impact image quality and detection accuracy. By presenting innovative approaches to optimize image capture processes, including techniques for handling occlusions, low-light environments, and minimizing processing latency, this paper will highlight potential breakthroughs that can be applied to real-world vehicle environments.

Additionally, the research will consider the balance between computational efficiency and detection accuracy, with the aim of developing a system that can operate seamlessly in real-time, without compromising performance or usability. Ultimately, this work aims to contribute to the broader field of intelligent transportation systems, offering insights into how improved image capture techniques can lead to more reliable and proactive drowsiness detection, thereby enhancing overall road safety.

II. Overall system Algorithm

Below is the way how the drowsiness detection system works. The drowsiness detection system operates as follows: The driver initiates the system by pressing the "Start Assist" button. Once activated, the system continuously monitors the driver's condition by capturing images every 0.25 seconds using a camera mounted in front of the driver. These images are then sent to a database for processing by an algorithm.

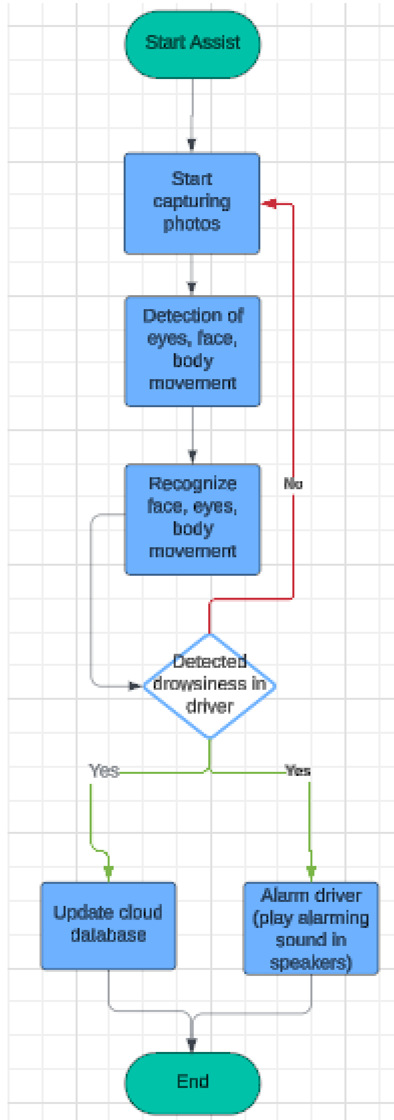


Figure 1: Overall system algorithm

The camera captures rapid images of the driver's eyes, face, and the road. The algorithm analyzes these images to assess the driver's state. It specifically monitors the driver's head movements and checks if the eyes remain within a predefined threshold to determine alertness. Additionally, a the backward-facing camera captures images of the road to ensure the driver is not swerving and driving safe.

The dataset used by the algorithm includes images of eyes, faces, and road conditions. By evaluating both the driver's physical state and driving behavior, the system can detect signs of drowsiness and ensure the driver is operating the vehicle safely

III. Methodology

III.1 OpenCV: Overview and features

OpenCV(Open Source Computer Vision library) – **OpenCV** is an open-source library which is used for real-time computer vision applications. It provides a wide set of tools for image processing, feature detection, and image analysis, making it ideal for developing a driver drowsiness detection system.

The techniques used in this paper were helped by the **OpenCV** library of python by using the methods below:

- **Image Processing:** Functions for manipulating and enhancing images, including filtering, edge detection, and morphological transformations. Utilizing functions such as **cv2.filter2D()** for filtering, **cv2.Canny()** for edge detection, and **cv2.morphologyEx()** for morphological transformations
- **Feature Detection and Matching:** Methods to detect and match features, such as facial landmarks, crucial for identifying signs of drowsiness. Employing methods like `cv2.detectMultiScale()` for detecting faces and `cv2.matchTemplate()` for matching features, crucial for identifying signs of drowsiness.

- **Object Detection:** Algorithms for detecting objects within images, such as the **Haar Cascade** classifier for face and eye detection.

IV. System implementation

IV.1 Image caputuring

The capuring of the images would be done by a camera no lower than 8-megapixels. Access to the camera inside the vehicle is made possible by cv2.VideoCapture(0).

IV.2 Image processing

IV.2.1 Haar Cascade Algorithm

The technique of this system for face detection and recognition will be the Haar Cascade. The reason why this algorithm is one of the best choices is that it can run in real time.
Haar cascade uses the cascading window, and it tries to compute features in every window and classify whether it could be an object. This method will use the light and shadow to detect objects in images.
Features:
The object detection procedure relies on the extraction of simple features from images. Unlike pixel-based methods, features provide a more efficient representation of visual patterns and encode domain knowledge critical for effective classification. The rectangles the features will be using, refer to regions of pixel data within an image. Note: the rectangles will be used in both vertical and horizontal directions to ensure precise calculation.

- **Two-Rectangle Feature:** This feature computes the difference between the sum of pixels within two horizontally or vertically adjacent rectangular regions (see Figure 1).
- **Three-Rectangle Feature:** Here, the sum within two outside rectangles is subtracted from the sum within a center rectangle (see Figure 2).
- **Four-Rectangle Feature:** This feature calculates the difference between diagonal pairs of rectangles (see Figure 3).

Figure 2: Two-Rectangle feature

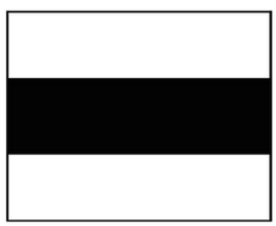


Figure 3: Three-Rectangle feature

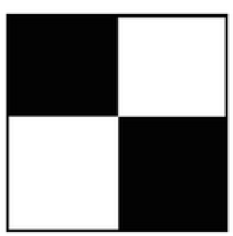


Figure 4: Four-rectangle feature

IV.2.2 Integral Image for Fast Computation

An intermediate representation known as the integral image will be used to efficiently compute rectangle features. Using the integral image makes it easy for us to carry out calculations quickly on many features simultaneously, allowing functions such as detecting if a driver is falling asleep to be done fast.
The integral image at location x,y is the sum of pixels above x,y and to the left of x,y.

$$ii(x,y) = \sum_{x' \leq x, y' \leq y} i(x', y')$$

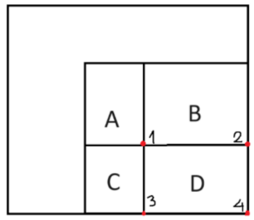


Figure 5: Four location feature

The sum of the pixels within rectangle D, can be computed with four array references. The value of the integral image at location 1 is the sum of pixels in A-rectangle. The value of location 2 is sum of pixels in B and A rectangle. At location 3 is sum of rectangles A and C, and at 4 is sum of pixels in all (A, B, C, D) rectangles. The sum in rectangle D can be computed in the following way (1+4) – (2+3).

After computing integral images, the algorithm utilizes them to efficiently calculate rectangular features required for object detection. The process involves feature calculation, where given a rectangular region of interest, the algorithm retrieves pixel sums from the integral image, representing the cumulative intensity values within the region. Using these pixel sums, the algorithm computes desired rectangular features, such as Haar-like features, important for distinguishing between object and non-object regions. Once the features are computed, they are fed into a classifier, such as a boosted cascade of weak classifiers, for object detection. The classifier evaluates the features and makes a decision about whether the region contains the object of interest based on learned criteria. In the case of cascade classifiers, multiple stages of feature evaluation are employed, each with increasing complexity. Regions that pass through one stage are further evaluated by subsequent stages, while regions that fail to meet certain criteria are quickly discarded, optimizing computational efficiency.

IV.2.3 Face Recognition

Localizing the detected face will be another complicated process. The detection of face attributes will be a subset of picture shape predictors. The important facial attributes will be:

- Eyes
- Eyebrows
- Lips
- Nose
- Jawline

After localizing the points for each face attribute, we will be able to exctract the regions in the face with the help of the landmarks (points around the facial features, which outline the shape of the feature) by using the shape predictor method.

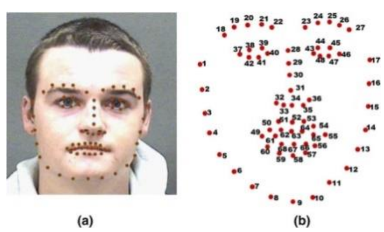


Figure 6: Landmarks over facial features

IV.2.4 EAR (Eye Aspect Ratio) evaluation and yawn detection

The EAR is a metric that determines eye openness or closeness. The system uses the Eye Aspect Ratio to determine the states which the driver will go through, thus alarming him if he closes his eyes. The EAR is computed using the Euclidean distance in a vertical and horizontal direction between the upper and lower eyelids. The EAR also has an evaluation function, where an open eye has a high EAR than the closed eye, hence the algorithm will try to maximize this value. Another technique the system will use is PERCLOS. The PERCELOS (percentage of eye closure) metric is a fatigue detection metric that calculates eye closure rate over the pupil in a given time frame. It is established on gradual and slower eyelid closures or eyelid sliding rather than regular eye blinks. This approach can ignore regular eye blinks and precisely detect sluggish eyelid closures or eyelid slides caused by fatigue driving.

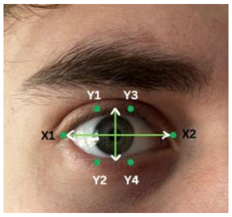


Figure 7: Open eye

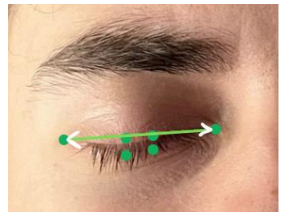


Figure 8: Closed eye

According to the points in figure 6 and 7, the vertical distance of the eye is calculated as:
Verticaldistance = (LY1 – LY2) + (LY3 – LY4).
The LY1 and other variables denoted by L mean the corresponding coordinates of that point in the image.
Horizontaldistance = LX1 + LX2
With the help of the vertical and horizontal distance the EAR will be calculated as:

$$EAR = \frac{\text{Verticaldistance}}{2*\text{Horizontaldistance}}$$

After calculating the EAR from the images captured by the device, the next step involves detecting whether the driver is actually drowsy or not. Firstly a threashhold needs to be set in order to make the correct class labeling of the images. We need to train the model by posing as drowsy and taking the pictures for the training set. And after calculating the threashold, we can use it to determine the drowsiness for other images.

IV.2.5 Yawn detection:

Yawning is also one of the main causes of drowsiness. Detecting it in a precise and correct manner is essential for the system. The calculation of the yawn will happen by getting the distance of the landmarks on the drivers lips before and after the yawn.

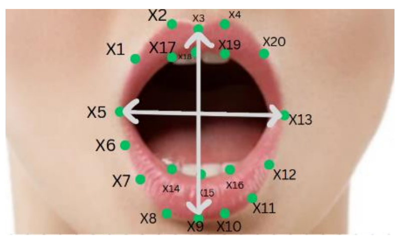


Figure 9: Top and lower lip points

The distance of the top and lower lip points is calculated by subtracting the mean of the top lip weights (coordinates) with the points on his lower lip. The distance of the upper lip is calculated as:

$$TopLip = \sum_{i=2}^{4} Lxi + \sum_{i=17}^{19} Lxi$$

And the lower lip:

$$LowerLip = \sum_{i=14}^{16} Lxi + \sum_{i=8}^{10} Lxi$$

The total lip distance between the lower and upper lip is calculated as follows:

$$LipDistance = \frac{TopLip - LowerLip}{8}$$

V. Road analysis

In addition to face recognition the importance of road analysis is essential. This component focuses on detecting lane markings and whether the driver is maintaining a proper lane discipline. The process for lane detection and tracking is done in several steps: capturing images, enhancing the images for a more precise analysis and detection, identifying lane markings using edge detection and Hough Transform, and monitoring the vehicles position relative to the marked lanes.

V.1 Image preprocessing

Given the fact that road colors can be two opposite, light and dark, for a more simple processing we can turn the images into greyscale. Other than turning the image into black and white smoothing it out and reducing the noise is also essential. With the help of OpenCV functions an example of the converted image will be as follows:

```
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
blur = cv2.GaussianBlur(gray, (5, 5), 0)
```

Methods of simplifying the image – **Canny edge detector**:

- **Non-maximum suppressison**:
  - Non-maximum suppression is used to thin the detected edges. It works in the way that it only keeps the strongest edge points and removes the others. For every pixel in the image and its magnitute is compares with those of its neighbours along that direction. If a pixel isn't a local maximum its value is set to zero.
- **Double threasholding**:
  - The threasholding will be used to classify the strength of the pixels. Two threasholds will be kept, a low and a high threashold. If a pixel's gradient magnitude is above the high threashold it will be considered as a strong edge, those between the two threashold will be considered weak, and those below the low will be considered as supressed.
- **Edge tracking by Hysteresis**:
  - Hysteresis is a technique used in edge detection, notably in algorithms like Canny edge detection. It employs two thresholds: a high threshold and a low threshold.

    1. **Strong Edges**: Pixels with gradient values above the high threshold are classified as strong edges, representing definitive boundaries in the image.

    2. **Weak Edges**: Pixels with gradient values between the low and high thresholds are labeled as weak edges. These are only considered edges if they are connected to strong edges.

    3. **Noise Reduction**: This method effectively minimizes noise by filtering out weak edges that aren't linked to strong ones, preserving important features while eliminating less relevant details.

    Overall, hysteresis enhances edge representation in images, making it a valuable tool in computer vision and image processing.
  - Hysteresis is a technique, where it filters the pixels in a way that it keeps only the strong edges, while reducing noise. It keeps the weak edges if and only if they are directly connected to strong edges.

After detecting the edges the lines of the road need to be detected too. The Hough Transform will help map the edges in the image space to curves in the Hough space.

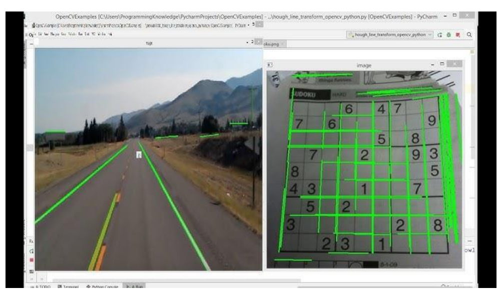


Figure 10: Line detection with Hough Transform

VI. Conclusion

By integrating road analysis with drowsiness detection, this system presents a brilliant approach to enhancing driving safety. Through the monitoring of both driver condition and lane swerving, it offers a solution for completely avoiding the risks associated with drowsy driving. This approach not only alerts drivers to their own fatigue but also alerts drivers to maintain lane discipline.
Integrating road analysis with drowsiness detection can greatly enhance driving safety through several important features:
1**. Real-Time Monitoring**: The system continuously evaluates the driver's alertness by analyzing metrics such as eye movement, head position, and reaction times. This enables early identification of drowsiness before it leads to dangerous situations.
2. **Lane Detection**: By assessing lane markings and vehicle positioning, the system can detect swerving or unintentional lane departures. If the vehicle drifts out of its lane, the system will issue alerts to the driver.
3. **Alerts and Interventions**: When signs of drowsiness are detected, the system can provide visual or auditory alerts to prompt the driver to take a break
4. **Data Integration**: The system can combine data from various sources, such as traffic conditions and weather, to provide context-aware alerts. For instance, if it detects heavy traffic along with signs of fatigue, it may recommend taking a break.
5**.User-Friendly Interface**: An intuitive interface ensures that alerts are clear and not overly disruptive. Visual cues can be shown on the dashboard, while auditory alerts can effectively grab the driver's attention without being too intrusive.
6. **Long-Term Benefits:** Over time, the system can gather data on driving habits and fatigue patterns, offering personalized feedback and recommendations. This can help drivers improve their habits and lower the risk of drowsy driving incidents.
7. **Potential Integration with Autonomous Systems**: As autonomous driving technology evolves, this drowsiness detection system could be integrated into semi-autonomous vehicles, allowing for smoother transitions between manual and automated driving, further

enhancing safety.

By addressing both driver alertness and road awareness, this integrated approach aims to significantly reduce accidents caused by drowsy driving, ultimately making roads safer for everyone.

VII. References

[1] National Highway Traffic Safety Administration (NHTSA). "Drowsy Driving and Automobile Crashes." [Online]. Available: NHTSA.

[2] OpenCV Documentation. "Open Source Computer Vision Library." [Online]. Available: OpenCV.

[3] P. Viola and M. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features," Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001.

[4] F. Bergasa et al., "Real-time system for monitoring driver vigilance," IEEE Transactions on Intelligent Transportation Systems, vol. 7, no. 1, pp. 63-77, March 2006

[5] G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955. (references)

[6] Fleuret and D. Geman. Coarse-to-fine face detection. Int. J. Computer Vision, 2001.

[7] William T. Freeman and Edward H. Adelson. The design and use of steerable filters. IEEE Transactions on Pattern Analysis and Machine Intelligence

Journal Of Computer Sciences and Mathematics – JCSM

ISSN 2955-246X

Volume 1, Issue2, 2025