AI-Powered Pipeline for Video Generation from Large Texts

1st Andrej Nankov
MSc Candidate, Software Engineer
Faculty of Computer Science and Engineering
Skopje, N.Macedonia
andrej.nankov@students.finki.ukim.mk

2th Vladimir Trajkovik
PhD, professor
Faculty of Computer Science and Engineering
Skopje, N.Macedonia
trvlado@finki.ukim.mk

***Abstract*—The demand for video content is growing fast, driven by AI advancements, the high-speed Internet, and monetization on platforms such as YouTube and TikTok. Generating high-quality videos from large textual formats, such as books and long-form articles, remains challenging due to contextual inconsistencies and information loss. This paper presents a structured AI-powered processing pipeline designed to transform extensive textual content into video materials while preserving coherence and key information. The pipeline consists of four key stages: (1) Text Summarization and Keyword Extraction, utilizing the TextRank algorithm to generate concise and meaningful content; (2) Configuration Planning, where video elements are structured in JSON format for seamless AI-based processing; (3) AI-driven content Generation, integrating ChatGPT for script development and InvideoAI for automated video production; and (4) Post-processing and User Refinement, ensuring adaptability and editability of the final output.**

The proposed method is evaluated in practice using both article-based and book-chapter inputs, demonstrating its effectiveness in generating contextually relevant and structured video content. Experimental results indicate that the pipeline successfully reduces manual intervention and improves processing efficiency, though challenges remain in handling non-textual elements such as images, graphs, and formulas. Future work will explore enhanced AI summarization techniques, multi-modal content integration, and real-time video generation improvements. This research contributes to the growing field of AI-assisted content creation by providing a scalable and adaptable approach to transforming large-scale textual data into engaging video materials.

***Index Terms*—AI, Text Summarization, Video Processing Pipeline, Content Generation, Media Transformation.**

I. Introduction

In the era of digital transformation, the demand for automated content generation increases driven by advances in artificial intelligence (AI), natural language processing (NLP), and video production technologies. The ability to convert large-scale textual content, such as books, research papers, and long-form articles, into engaging video material has significant applications in education, media, and digital marketing. Existing AI-driven video generation tools often struggle with context retention, coherence, and scalability [1] [2], which leads to the need for a more structured and efficient processing pipeline. Advancements in artificial intelligence (AI) have led to the rapid development of tools such as Microsoft Copilot, Google Gemini, Meta AI, and ChatGPT, which are widely used for reasoning-based tasks [3]. Among recent AI-based tools, DeepSeek has introduced DeepSeek-R1 [4], a reasoning model built using reinforcement learning (RL) without supervised fine-tuning. This model demonstrates remarkable reasoning abilities in benchmark evaluations, effectively addressing challenges such as poor readability and language inconsistencies. A key advantage of DeepSeek-R1 is its ability to operate on lower computing power, making it suitable for self-hosted applications and expanding the possibilities for AI deployment. Currently, DeepSeek-R1 processes only textual data, but it holds significant potential for accelerating AI-driven content processing pipelines, particularly for tasks involving large-scale text analysis and transformation. Benchmark results indicate that DeepSeek-R1 achieves performance levels comparable to premium models like ChatGPT-4o [4], making it an attractive alternative for cost-effective AI integration. While ChatGPT remains advantageous due to its easy API integration and affordability under subscription-based models, DeepSeek-R1 offers greater flexibility for self-hosted AI solutions [5]. This research presents an AI-powered workflow designed to transform extensive textual data into structured video narratives, leveraging text summarization, keyword extraction, structured configuration planning, and AI-assisted video synthesis. By addressing the challenges of automated text-to-video conversion, this study contributes to enhancing digital content accessibility, optimizing AI-driven storytelling, and improving the scalability of video content production. This paper is structured as follows: Section II provides background information for the paper. Section III reviews existing data processing frameworks discussing their relevance to AI-driven video creation. Section IV details the proposed processing pipeline, outlining its core stages. Section V presents experimental results, evaluates the pipeline and performance using articles and book chapters, and discusses key challenges. Section VI proposes potential enhancements in multi-modal AI processing and real-time video generation. Finally, Section VII concludes the paper.

II. Background

This research is inspired by two key previous projects: a multi-tenant e-learning platform and a SaaS-based video editing platform. Both projects are fully developed, production-ready solutions that have provided valuable insights into scalability, automation, and processing pipelines, which form the foundation of this study.

A. Multi-Tenant E-Learning Platform

One of the biggest challenges in e-learning platforms is ensuring efficient video availability for course materials. The proposed solution incorporates RabbitMQ Stream, decentralized object storage, and Content Delivery Networks (CDN) to enhance video distribution, scalability, and system responsiveness. The multi-tenant architecture ensures that multiple users and organizations can access the platform simultaneously while maintaining performance and data isolation. This experience has shaped the research by emphasizing the need for automated, scalable, and flexible video content management.

B. SaaS-Based Video Editing Platform

The second project focuses on video editing in a SaaS environment, where the video timeline creation process follows a structured pipeline. Each stage in this pipeline is responsible for a specific function, ensuring an organized workflow for video processing and rendering. The platform supports essential editing actions such as merging, cutting, muting, ordering, and drag-and-drop operations.

A key challenge on this platform was to handle longer video formats while maintaining efficient execution. To address this, Laravel Horizon was integrated, enabling asynchronous execution, job queueing, and real-time performance monitoring within the Laravel ecosystem. This system played a crucial role in understanding the complexities of video processing pipelines, which directly influenced the research presented in this paper.

C. Data Pipelines and Their Role in AI Processing

Data pipelines play a fundamental role in automating data transformation, enrichment, and processing before applying AI models. A typical pipeline consists of:

1) A Source - the raw input data, such as text, video, or structured datasets.
2) Processing Steps - A series of transformations, such as filtering, grouping, summarization, and AI-based analysis.
3) A Destination - The final structured output, ready for storage or further application.

For example, a single social media comment can be processed through different pipelines for real-time monitoring, sentiment analysis, or location-based mapping. Each application uses a distinct pipeline that must function seamlessly and efficiently.

Pipelines are widely used in event-driven architectures, enabling real-time processing while optimizing computing costs. The approach used in this research follows a batch processing model, where large textual data are processed asynchronously in stages. This method ensures scalability while allowing AI models to generate contextually accurate and structured video content.

III. Related work

Traditional data processing frameworks, such as MapReduce and Spark, have transformed the way large-scale textual and semi-structured data are handled [6]. However, video processing presents unique challenges due to the complexity and size of video files. Addressing these challenges requires scalable and efficient solutions that optimize resource usage while maintaining low latency and high processing accuracy.

One tool designed to tackle these challenges is Sprocket [6], a serverless video processing framework that aims to push the boundaries of cloud-based video transformation by offering an optimized, low-cost solution. Unlike text-based data pipelines, Sprocket is specifically designed to process video files and operates on a serverless cloud infrastructure.

Sprocket follows a modular pipeline model, breaking video processing tasks into distinct stages, such as decoding, filtering, and encoding. The framework leverages AWS Lambda to achieve dynamic scaling and efficient resource allocation. By distributing tasks across thousands of Lambda instances, Sprocket achieves high parallelism and low latency. The core benefits of Sprocket include:

1) Modularity - Developers can customize processing pipelines by composing multiple stages.
2) Scalability - The serverless architecture enables efficient handling of large-scale video processing tasks.
3) Cost-Effectiveness - The pay-as-you-go model minimizes costs while maintaining high-performance efficiency.
4) AI Integration – Sprocket can incorporate AI-based video analysis tools, such as Amazon Rekognition, to perform advanced tasks such as facial recognition and object detection.

Another notable example of data pipeline optimization is LinkedIn's real-time activity data processing system [7]. Managing real-time data streams presents significant challenges, particularly in advertising, search, recommendation systems, and security. LinkedIn's real-time pipeline is built around Apache Kafka, handling over 10 billion message writes daily across multiple subscribing systems.

Initially, LinkedIn relied on separate batch-oriented systems for user activity data and server metrics, which lacked real-time data access [7]. The inefficiencies in these legacy systems led to difficulties in correlating operational issues with business metrics. While LinkedIn experimented with off-the-shelf messaging solutions like ActiveMQ, performance issues under production loads led to the development of Kafka, a high-throughput, low-overhead data infrastructure.

Kafka functions as a write-ahead log, allowing subscribers to read and process data changes efficiently. Its design principles include:

- Data Partitioning - Distributing data across multiple servers to enhance scalability.
- Batch Processing and Data Compression - Improving performance while reducing network and file system overhead.

- Schema Management - Using Avro for data serialization to ensure compatibility and integration across multiple applications.

Kafka supports over 10 billion daily message writes, with sustained throughput exceeding 172,000 messages per second across 367 topics [7]. LinkedIn integrates Kafka with Hadoop, automatically loading data and structuring it into Hive tables for efficient batch processing [7].

A. Security Considerations in AI-Powered Pipelines

Ensuring the security and integrity of AI-driven pipelines is crucial, particularly when handling sensitive data. Vulnerabilities in AI models, such as data exposure, adversarial manipulation, and privacy risks, must be addressed [8]. Access control mechanisms, data encryption, and continuous monitoring are necessary to protect against security threats [8]. When integrating general AI models, it is essential to pre-filter sensitive data before sharing it with external AI tools. Additionally, ensuring compliance with data protection laws and AI ethics guidelines is necessary to prevent misuse or unauthorized data access [9].

IV. Processing pipeline framework

The proposed processing pipeline (Figure 1) follows the divide-and-conquer principle, where a long textual input (e.g., a book or article) is broken into manageable sections. Each section undergoes a structured pipeline, ensuring that essential content is extracted, transformed, and converted into video format while maintaining coherence and relevance. This modular approach allows efficient processing, where multiple short videos can be generated from different sections and later merged into a cohesive final video product. This strategy is beneficial for individual content creators, businesses, and educational institutions that require structured and contextually accurate video content while minimizing manual editing efforts.

The core pillars of this research are Data, AI, Transformation, and Execution, each corresponding to a key stage in the pipeline. The workflow consists of the following stages:

A. Initialization

To implement the processing pipeline, we employed Node.js along with a selection of built-in and third-party packages. Node.js was chosen due to its scalability, extensive community support, and widespread adoption in modern web and backend development. The pipeline operates primarily in a console-based environment, leveraging the built-in file system module to run multiple pipelines in parallel, thereby optimizing the time required to generate multiple videos per book or document. This stage is responsible for preparing the initial data for the next stages. For instance, read from a file, prepare the text from the defined sections, and remove unused empty spaces and strange characters over the text.

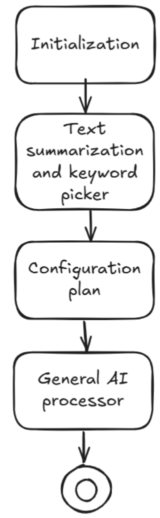

Fig. 1. Processing pipeline stages

B. Text Summarization and Keyword Extraction

The second stage involves text summarization [10] and keyword extraction using a node summarizer, which implements the TextRank algorithm. This algorithm is particularly effective for summarizing text from books, newspaper articles, and research papers by identifying key sentences and ranking them based on their relevance [11].

Summarization Process:

1) Sentence Tokenization – The text is split into individual sentences.
2) Preprocessing - Punctuation is removed, and all text is converted to lowercase.
3) Keyword Mapping - Each sentence is mapped to an array of nouns and adjectives.
    - Example: "The San Antonio is a good basketball team." will be taken as [San Antonio, good, basketball, team].
4) Graph-Based Ranking - A weighted graph is constructed where:
    - Each node represents a sentence.
    - Edges are formed between sentences based on shared nouns and adjectives.
    - Edge weights are determined by the frequency of shared words.
5) Random Walk Algorithm - Sentences are iteratively ranked based on their connectivity, with higher-ranked sentences forming the final summary.

This process ensures that the generated summaries retain essential information while eliminating redundancy.

C. Configuration planning

The third stage defines the video structure and behavior using JSON-based configurations. JSON was selected due to its lightweight nature, human readability, and compatibility with multiple programming languages, making it ideal for structured data exchange.

The configuration file specifies:

1) Video metadata (e.g., title, description).
2) Scene elements, including:
   - Type of visual effects.
   - Selection of backgrounds and colors.
   - Image overlays for enhanced representation.
   - A supportive narrator to provide voiceovers.
3) Audio parameters:
   - Background music selection.
   - Narrator attributes (tone, accent, gender).

Once finalized, this configuration is attached to the summarized text and passed to the AI-powered video generation stage. See Figure 2 for an example JSON configuration.

D. *Configuration planning*

The final stage leverages AI models to convert the processed content into fully structured video material. This is achieved through two API-based interactions:

1) ChatGPT API (GPT-3.5-Turbo, GPT-4-Turbo) – Generates a cohesive and engaging narration.
2) InvideoAI API – Automates video creation and rendering, integrating text, images, and narration into a structured video format.

Once the request is processed, the InvideoAI dashboard provides:

- A progress bar indicating the rendering status.
- Options for previewing and editing specific sections.
- Download capabilities for high-resolution video output (1080p at 60fps).

Through this approach, users gain greater control and flexibility, ensuring high-quality, AI-generated video content without losing contextual accuracy.

V. Evaluation of the Processing Pipeline

The processing pipeline was tested using different articles and book chapters to evaluate its effectiveness in generating contextually relevant videos. We can summarize that (1) The initial stage operated almost identically over different long texts according to time-consummation and processing power. (2) The text summarization and keywording stage is computationally lightweight, making it ideal for summarization and there is no need for pre-trained data. (3) The configuration plan directly depends on what we added to the configuration JSON format. Usually, this configuration is persistent but it offers flexibility to make the changes before we jump to the next stage. (4) The general AI processor is the most time-consuming stage since depends on making HTTP calls and communicating first with the API of ChatGPT and then InvideoAI. This stage is also subscription-based and depends directly on what type of plans we have picked for the ChatGPT model and InvideoAI plan. By opting for higher-tier plans, we can enhance video quality and the AI leverages the training data. Additionally, premium plans eliminate any restrictions on video length within a single pipeline run.


```json
{
  "title": "Habit Formation: The Power of Repetition",
  "description": "Exploring the transformative process of habit formation through
repetition and practice.",
  "scenes": [
    {
      "scene_id": 1,
      "visuals": {
        "type": "image",
        "source": "Illustration of neural pathways forming with repetition"
      },
      "narration": "Habit formation is a remarkable journey where behaviors
transition from conscious actions to automatic responses through repetitive
practice."
    },
    {
      "scene_id": 2,
      "visuals": {
        "type": "text_overlay",
        "elements": [
          "Automaticity",
          "Repetition",
          "Behavior Change"
        ],
        "background_type": "abstract",
        "background_color": "green"
      },
      "narration": "The duration to solidify a habit varies; what truly matters is
the consistent repetition that engrains the behavior into our daily routines."
    },
    {
      "scene_id": 3,
      "visuals": {
        "type": "text_list",
        "elements": [
          "Practice",
          "Consistency",
          "Transformation"
        ],
        "background_type": "minimal"
      },
      "narration": "Building a habit is less about the time it takes and more about
the dedication to practice. With persistence, habits evolve from intentional efforts
to subconscious actions."
    },
    {
      "scene_id": 4,
      "visuals": {
        "type": "image_collage",
        "images": [
          "Image of a person repeating a behavior",
          "Brain scan showing habit formation",
          "Illustration of habit becoming automatic"
        ]
      },
      "narration": "Repetition is key in habit formation. Through consistent
practice, behaviors shift from requiring attention to seamlessly integrating into
our daily lives."
    },
    {
      "scene_id": 5,
      "visuals": {
        "type": "text_overlay",
        "elements": [
          "Habit Formation",
          "Practice",
          "Consistency"
        ],
        "background_type": "mindful_patterns"
      },
      "narration": "Embrace the journey of habit formation. Explore the power of
repetition, practice, and unwavering consistency in transforming behaviors into
automatic habits."
    }
  ],
  "audio": {
    "background_music": "upbeat_and_professional",
    "narration_voice": {
      "gender": "male",
      "tone": "friendly",
      "accent": "neutral"
    }
  }
}
```


Fig. 2. Example JSON configuration

A. *Video Generation from Article*

InvideoAI played a crucial role in enhancing video quality by utilizing its built-in keyword-based video storage, which allowed seamless integration of abstract video elements into the generated content. The configuration stage (Fig. 2) performed exceptionally well, enabling automated scene selection based on summarized text, keywords, narration, and background audio metadata. One of the test articles focused on the software industry downturn related to the global economy, COVID-19, and war [12]. The pipeline successfully processed this content through all stages:

- Stage 1 (Initialization): The text was structured and preprocessed.
- Stage 2 (Summarization and Keyword Extraction): The TextRank algorithm effectively identified key terms,

including economy, COVID-19, and war, forming a weighted graph to ensure contextual relevance.
- Stage 3 (Configuration Planning): No modifications were required in this stage, as the predefined configuration effectively structured the video.
- Stage 4 (AI-Powered Video Generation):
  - ChatGPT generated narration scripts based on extracted keywords.
  - InvideoAI produced over 20 video scenes, each featuring distinct backgrounds, keywords, and visual representations (e.g., green screens, minimalistic visuals, and source explanations).
  - The background music was configured to be dramatic with a positive ending, aligning with the article's theme.
  - The narrator was male with a deep English voice, as per the content creator's preferences.

The final video had a duration of 2 minutes and 41 seconds [13] and was generated using the InvideoAI free plan. The tone and structure were preserved, ensuring that the generated content accurately represented the intent of the article.

B. Video Generation from Book

The second test involved summarizing a book chapter from Atomic Habits by James Clear [14]. The selected chapter, "Walk Slowly, but Never Backward", discusses the long-term impact of habit formation and the power of small changes over time.

- Stage 1 and Stage 2 (Initialization and Summarization):
  - The summarization model effectively extracted key takeaways, focusing on the role of small improvements in long-term success.
  - A graph image within the chapter was ignored, as it contained non-textual data. (This limitation is discussed further in the Domain Challenges section.).
- Stage 3 (Configuration Planning):
  - The same configuration template as used for articles was applied, ensuring consistency in narration, visuals, and background audio.
- Stage 4 (AI-Powered Video Generation):
  - The AI extended scene durations, adding more descriptive narration and additional visual effects to illustrate key concepts.
  - Selected images represented brain connections, behavioral repetition, and habit formation to complement the narration.
  - The background music was calming and study-oriented to match the book's self-improvement theme.
  - The narrator was male, with a friendly tone and a standard English accent.

The final video was 52 seconds [15], shorter than the article-based video but focused on delivering precise, structured content. While all textual elements were well-covered, the AI was unable to process and represent numerical data from graphs, indicating a potential limitation in handling non-textual elements.

C. Challenges and Limitations

While the proposed pipeline demonstrates scalability, several challenges must be addressed to enhance performance, cost efficiency, and adaptability.

One major limitation concerns scalability and performance bottlenecks. Although the system supports parallel execution of multiple processing pipelines, increasing throughput by adding more Node.js workers introduces computational complexity. Currently, the pipeline is limited to processing one long text at a time, and when multiple video creations are queued in InvideoAI, delays occur regarding asynchronous processing and load complexity, affecting real-time InvideoAI actions. A potential solution involves implementing a new pipeline stage before the AI processor stage which has dynamic worker allocation based on load-balancing techniques, which covers the asynchronous processing and optimizes resources for load complexity. For example, Pipeflow can be added as a stage to solve this problem [16].

Another critical challenge is the cost of cloud computing. Running the pipeline on cloud-based servers requires significant computational resources, and as the number of workers increases to enhance processing speed, operational costs rise. To address this, cost-efficient cloud strategies, such as serverless computing and AI model quantization, should be explored to reduce expenses while maintaining performance.

A notable limitation is the inability to process non-textual data, such as images, graphs, and mathematical formulas. Since the pipeline primarily relies on text-based transformations, it fails to interpret visual content, leading to information loss, especially in academic materials. A potential solution involves integrating image-processing libraries like OpenCV or Tesseract or using multimodal AI models capable of jointly analyzing text and visual data, which would enhance the system's ability to extract and summarize graphical content [17]. Efficiency in AI processing time is another area that requires improvement. While the preprocessing of multiple book chapters takes less than three seconds demonstrating high efficiency in summarization and keyword extraction, the most significant delay occurs during API communication with ChatGPT and InvideoAI, particularly during scene rendering and video generation. One way to mitigate this is by deploying local AI models, such as running open-source AI models on-premises instead of using ChatGPT models, which would reduce dependency on external APIs, lower latency, and improve overall system responsiveness [18].

VI. Future Improvements and Research Directions

To further enhance the pipeline's usability and effectiveness, several areas of future research can be proposed. Optimizing the data pipeline for video merging would allow multiple generated videos to be combined into a single long-form output, such as full-length educational content, or tuning AI models, such as GPT-3.5 or GPT-4, for domain-specific summarization could enhance content relevance and reduce unnecessary information, with the possible incorporation of knowledge-graph-based summarization techniques for handling complex subjects [19]. Advancing the pipeline's ability to process both text and images is another key area for improvement. By integrating Vision Transformers (ViTs)[20] and OCR-based solutions[21], the system could interpret charts, graphs, and mathematical formulas, addressing one of the primary limitations identified in the current approach. Moreover, enhancing AI-based speech synthesis by incorporating fine-tuned speech models would allow for adaptive tone modulation based on content type, making the generated videos more engaging and suitable for various applications, including academic content, storytelling, and marketing materials [22].

Cost optimization strategies must be explored to make the pipeline more sustainable. Implementing low-latency alternatives to commercial AI services—such as local inference models or open-source AI alternatives-could significantly reduce reliance on costly third-party APIs while maintaining high-quality video generation.

VII. Conclusion

This paper presents an AI-powered processing pipeline designed to transform large textual content into structured video materials. Using text summarization, keyword extraction, structured configuration planning, and AI-driven video generation, the proposed pipeline converts books and long-form articles into contextually accurate videos while minimizing manual intervention.

Experimental results indicate that the pipeline produces structured video outputs that align with the original textual intent. The use of ChatGPT and InvideoAI ensures natural language processing capabilities, allowing cohesive narration and visual storytelling. Limitations in handling non-textual data, such as images, graphs, and mathematical formulas, were observed requiring further enhancements for multimodal content processing.

Future research will focus on optimizing the AI models for improved content relevance and summarization accuracy, integrating Vision Transformers (ViTs) and OCR-based solutions to process non-textual elements, and developing cost-efficient AI alternatives to reduce reliance on commercial APIs. Additionally, advancements in speech synthesis and AI-driven tone modulation will further improve engagement and adaptability for different audiences.

References

[1] F. Bao, C. Xiang, G. Yue, G. He, H. Zhu, K. Zheng, M. Zhao, S. Liu, Y. Wang, and J. Zhu, "Vidu: a Highly Consistent, Dynamic and Skilled Text-to-Video Generator with Diffusion Models," arXiv, vol. 2405, no. 04233, pp. 1-45, May 2024. doi: 10.48550/arXiv.2405.04233.

[2] F. Waseem and M. Shahzad, "Video Is Worth a Thousand Images: Exploring the Latest Trends in Long Video Generation," arXiv, vol. 2412, no. 18688, pp. 1-35, Dec. 2024. doi: 10.48550/arXiv.2412.18688.

[3] F. Rosén and M. Rübner, "Augmented Large Language Models for Software Engineering," Master Thesis, Dept. of Design Sciences, Lund University, Lund, Sweden, 2024.

[4] DeepSeek-AI, "DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning," arXiv:2501.12948v1 [cs.CL], Jan. 2025.

[5] I. Solaiman, R. Bommasani, D. Hendrycks, A. Herbert-Voss, Y. Jernite, A. Skowron, and A. Trask, "Beyond Release: Access Considerations for Generative AI Systems," arXiv:2502.16701v1 [cs.CY], Feb. 2025.

[6] L. Ao, L. Izhikevich, G. M. Voelker, and G. Porter, "Sprocket: A Serverless Video Processing Framework," in Proceedings of ACM Symposium on Cloud Computing (SoCC '18), Carlsbad, CA, USA, October 11–13, 2018, pp. 263-274. doi: 10.1145/3267809.3267815.

[7] K. Goodhope, J. Koshy, J. Kreps, N. Narkhede, R. Park, J. Rao, and V. Y. Ye, "Building LinkedIn's Real-time Activity Data Pipeline," Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, 2012.

[8] H. I. Kure, P. Sarkar, A. B. Ndanusa, and A. O. Nwajana, "Detecting and Preventing Data Poisoning Attacks on AI Models," arXiv, vol. 2503, no. 09302, pp. 1-9, Mar. 2025. doi: 10.48550/arXiv.2503.09302.

[9] G. Feretzakis, K. Papaspyridis, A. Gkoulalas-Divanis, and V. S. Verykios, "Privacy-Preserving Techniques in Generative AI and Large Language Models: A Narrative Review," Information, vol. 15, no. 11, pp. 697, Nov. 2024. doi: 10.3390/info15110697.

[10] A. K. Yadav, A. K. Maurya, Ranvijay, and R. S. Yadav, "Extractive Text Summarization Using Recent Approaches: A Survey," Ingénierie des Systèmes d'Information, vol. 26, no. 1, pp. 109-121, Feb. 2021. doi: 10.18280/isi.260112.

[11] V. Gulati, D. Kumar, D. E. Popescu, and J. D. Hemanth, "Extractive Article Summarization Using Integrated TextRank and BM25+ Algorithm," in Proceedings of the International Conference on Advanced Computing and Communication Systems (ICACCS), 2023, pp. 1-6. doi: 10.1109/ICACCS.2023.123456.

[12] A. Nankov, "Software Industry Depression," Medium, 2023. [Online]. Available: https://medium.com/@nanorocks/software-industry-depression-51c69e0e0ed5. [Accessed: 15-Mar-2025].

[13] A. Nankov, "Video from Article - Software Industry Depression" GitHub repository, Available: https://github.com/nanorocks/created-videos-research-ai-pipeline/blob/main/SoftwareIndustryDepressionArticle.mp4. Accessed: Mar. 15, 2025.

[14] J. Clear, "Walk Slowly, But Never Backward," in Atomic Habits, New York, NY, USA: Penguin Random House, 2018.

[15] A. Nankov, "Video from Book Atomic Habits - Chapter 11" GitHub repository, Available: https://github.com/nanorocks/created-videos-research-ai-pipeline/blob/main/AtomicHabitsChapter11.mp4 Accessed: Mar. 15, 2025.

[16] C.-H. Chiu, T.-W. Huang, Z. Guo, and Y. Lin, "Pipeflow: An Efficient Task-Parallel Pipeline Programming Framework using Modern C++," arXiv, vol. 2202, no. 00717, pp. 1-13, Feb. 2022. doi: 10.48550/arXiv.2202.00717.

[17] S. Nazeema and S. G. Gundabatini, "Real-Time Text Detection and Translation Using OpenCV and Tesseract OCR: An Integrated Framework for Images and Videos," International Journal of Innovative Research in Technology, vol. 11, no. 6, pp. 2133-2139, Nov. 2024.

[18] J. Radas, B. Risse, and R. Vogl, "Building UniGPT: A Customizable On-Premise LLM-Solution for Universities," in Proceedings of EUNIS 2024 Annual Congress, Athens, Greece, 2025, pp. 108-116.

[19] M. Kalender, M. T. Eren, Z. Wu, O. Cirakman, S. Kutluk, G. Gultekin, and E. E. Korkmaz, "Videolization: Knowledge Graph Based Automated Video Generation from Web Content," Multimedia Tools and Applications, vol. 75, no. 24, pp. 4275-4294, Dec. 2016. doi: 10.1007/s11042-016-4275-4.

[20] M. Li, T. Lv, J. Chen, L. Cui, Y. Lu, D. Florencio, C. Zhang, Z. Li, and F. Wei, "TrOCR: Transformer-based Optical Character Recognition with Pre-trained Models," arXiv preprint arXiv:2109.10282, Sep. 2022.

[21] P. H. Jain, V. Kumar, J. Samuel, S. Singh, A. Mannepalli, and R. Anderson, "Artificially Intelligent Readers: An Adaptive Framework for Original Handwritten Numerical Digits Recognition with OCR Methods," Information, vol. 14, no. 6, pp. 305, May 2023. doi: 10.3390/info14060305.

[22] T.-Q. Lin, W.-P. Huang, H. Tang, and H.-Y. Lee, "Speech-FT: A Fine-tuning Strategy for Enhancing Speech Representation Models Without Compromising Generalization Ability," arXiv, vol. 2502, no. 12672, pp. 1-12, Feb. 2025. doi: 10.48550/arXiv.2502.12672.