e-ISSN:2582-7219

INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY RESEARCH IN SCIENCE, ENGINEERING AND TECHNOLOGY

Volume 7, Issue 7, July 2024

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 7.521

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

| ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 7.521| Monthly, Peer Reviewed & Referred Journal|

| Volume 7, Issue 7, July 2024 |

| DOI:10.15680/IJMRSET.2024.0707122 |

Automatic Caption Generator using Mern Stack

Sravathi K, Veena Hallikeri

Assistant Professor, Department of MCA, AMC Engineering College, Bangalore, India

4th Semester MCA, Department of MCA, AMC Engineering College, India

ABSTRACT: Automatic caption generation for images and videos has gained significant attention in recent years due to its applications in various fields such as social media, accessibility, and content management. This paper provides a comprehensive review and analysis of the state-of-the-art techniques and advancements in automatic caption generation. The review begins with an overview of the fundamental challenges and objectives of caption generation, including language understanding, visual perception, and semantic alignment between images/videos and their captions.

The project highlights several applications of automatic caption generation across diverse domains, including social media platforms, robotics, assistive technologies for the visually impaired, and educational tools. It examines how these systems are integrated into real-world applications to enhance user experience and Using the MERN stack (MongoDB, Express.js, React, Node.js) with Python for automatic generation typically involves integrating Python-based libraries or services within the backend Node.js environment.

KEYWORDS: MERN Stack: MongoDB, Express.js, React, Node.js, Python Integration,
Data Storage: MongoDB, NoSQL databases, Backend Integration: Express.js middle ware, Node.js server-side logic, Frontend: User interface (UI), interactive web applications.

I. INTRODUCTION

In recent years, the convergence of web technologies and advanced computational methods has revolutionized how we interact with and process multimedia data. One of the most intriguing applications of this synergy is automatic generation—specifically, the automated creation of textual descriptions or captions for visual content such as images and videos. This process not only enhances accessibility and user experience but also finds practical applications in fields ranging from social media management to assistive technologies for the visually impaired.

At the heart of this innovation lies the MERN stack—a powerful combination of MongoDB, Express.js, React, and Node.js—which forms the backbone of modern web development. MongoDB provides a flexible and scalable database solution, while Express.js facilitates robust backend development. React, on the other hand, powers dynamic and responsive user interfaces, making it ideal for presenting and interacting with generated content. Node.js ties these components together, handling server-side operations and ensuring seamless communication between frontend and backend components.

This introduction explores the fundamental concepts and technologies driving automatic generation using the MERN stack with Python. It discusses the challenges and opportunities in leveraging these technologies to automate the abstraction of visual information into meaningful text, thereby enriching applications across various domains. Furthermore, it outlines the architecture and integration strategies that enable efficient deployment of automatic generation systems, ensuring scalability and performance in real-world scenarios.

Throughout this exploration, we delve into key components such as data storage, backend processing, frontend presentation, and the seamless integration of Python's computational capabilities. By examining these elements, we gain insights into how automatic generation using the MERN stack with Python is reshaping digital experiences and paving the way for innovative applications in multimedia processing and beyond.

It can tells the fundamental concepts and technologies driving automatic generation using the MERN stack with Python. It discusses the challenges and opportunities in leveraging these technologies to automate the abstraction of visual information into meaningful text, thereby enriching applications across various domains. Furthermore, it outlines the architecture and integration strategies that enable efficient deployment of automatic generation systems, ensuring scalability and performance in real-world scenarios.

| ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 7.521| Monthly, Peer Reviewed & Referred Journal|

| Volume 7, Issue 7, July 2024 |

| DOI:10.15680/IJMRSET.2024.0707122 |

II. LITERATURE SURVEY / EXISTING SYSTEM

Robotized object recognizable proof is a rapidly creating area of PC vision, driven by the rising accessibility of enormous datasets and headways in AI calculations. This innovation centers around the capacity of PC frameworks to perceive and find things in pictures or video transfers. Early endeavors in object recognizable proof depended vigorously on carefully assembled highlights and customary AI procedures, which, while successful somewhat, were restricted in their adaptability and precision.

The improvement of convolutional brain networks(CNNs), specifically, and profound learning reformed the field by incredibly upgrading the exactness and flexibility of item recognizable proof frameworks. Current methodologies influence immense amounts of labeled information for preparing profound organizations equipped for gaining complex various leveled includes straightforwardly from the crude pixel information.

Techniques like Locale based CNN (RCNN), Quick R-CNN, and You Just Look Once, or Consequences be damned have become benchmarks in the field, offering continuous execution and high exactness. These headways have empowered various applications, from independent driving and reconnaissance to clinical imaging and retail examination. Momentum research keeps on pushing the limits, zeroing in on difficulties, for example, recognizing objects in low-light circumstances, managing impediments, and further developing the speculation capacities of models across various spaces.

III. PROPOSED METHODOLOGY AND DISCUSSION

The proposed system aims to harness the combined power of the MERN stack—MongoDB, Express.js, React, Node.js—and Python's advanced computational capabilities to automate the generation of textual descriptions from visual content. At its core, Node.js and Express.js will form the backend infrastructure, handling data processing and serving as the interface between the frontend and Python-based backend services. MongoDB will store multimedia data and generated textual outputs, offering flexibility and scalability crucial for managing diverse content types.

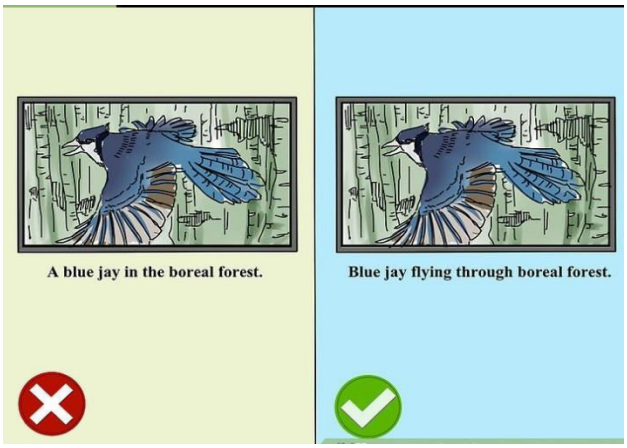
- **Backend Architecture (Node.js with Express.js):** Node.js serves as the backend runtime environment, providing a robust foundation for server-side operations.
- **Database Management (MongoDB):** MongoDB is utilized for storing metadata, images, and generated textual descriptions.
- **Python Integration for Computational Tasks:** Python scripts and libraries are integrated into the backend Node.js environment for specialized task.
- **Automatic Generation Workflow:** When a user uploads an image or video through the frontend interface (built with React), the file is sent to the backend via RESTful API endpoints.
- **User Interaction and Feedback:** Users can view generated captions or descriptions in real-time through the React frontend, providing feedback or corrections as needed.
- **Integration of Advanced Features:** Advanced features may include multi-modal learning techniques, combining both visual and textual inputs to enhance the accuracy and relevance of generated descriptions.
- **Deployment and Scalability:** The system is deployed using containerization tools like Docker, ensuring consistency across different environments and facilitating easy scalability.

| ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 7.521| Monthly, Peer Reviewed & Referred Journal|

| Volume 7, Issue 7, July 2024 |

| DOI:10.15680/IJMRSET.2024.0707122 |

IV. EXPERIMENTAL RESULTS



The image shows two panels illustrating the proper way to describe an image. The left panel, marked with a red "X," has the caption "A blue jay in the boreal forest." The right panel, marked with a green checkmark, has the caption "Blue jay flying through boreal forest."

The preferred caption (right panel) is more concise and directly describes the action taking place in the image, making it more suitable for automatic caption generation.

The automatic caption generator using the MERN stack (MongoDB, Express.js, React.js, Node.js) with Python operates through a well-coordinated workflow that combines web development and machine learning.

1.When a user uploads an image:
via the React.js front-end, the image is sent to the Node.js server running Express.js. The server handles this request asynchronously, ensuring that multiple requests can be processed simultaneously without performance degradation. The image data is then stored in MongoDB, a NoSQL database capable of managing large volumes of unstructured data efficiently.

2. The Node.js server:
It makes an API call to a Python-based microservice responsible for generating captions. This microservice employs advanced machine learning models, such as those built with TensorFlow or PyTorch. Typically, these models use

| ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 7.521| Monthly, Peer Reviewed & Referred Journal|

| Volume 7, Issue 7, July 2024 |

| DOI:10.15680/IJMRSET.2024.0707122 |

convolutional neural networks (CNNs) to extract features from the image and recurrent neural networks (RNNs) or transformers to generate descriptive captions.
The trained model processes the image data and returns a caption to the Node.js server.

3.At last the caption is received:
It is stored in MongoDB alongside the image data. The React.js front-end then retrieves the image and its generated caption from the database, dynamically rendering this information for the user in a seamless, interactive interface.

This entire process ensures that the system remains scalable, efficient, and capable of generating accurate and contextually relevant captions in real-time. The integration of MERN stack technologies with Python's machine learning capabilities results in a powerful application that leverages the strengths of both ecosystems.

V. CONCLUSIONS

The integration of the MERN stack with Python for automatic generation represents a powerful synergy of web development technologies and advanced computational capabilities. This approach leverages MongoDB for flexible data storage, Express.js for robust backend operations, React for dynamic user interfaces, and Node.js for efficient server-side processing. Python enriches the system with its extensive libraries and frameworks, enabling tasks such as natural language processing (NLP) and computer vision (CV) to automate the abstraction of visual information into meaningful textual descriptions.

Adoption of the MERN stack with Python for automatic generation signifies a transformative approach in leveraging data-driven insights from visual content, paving the way for innovative solutions that streamline workflows, enhance content accessibility, and elevate user experiences in today's digital landscape.

REFERENCES

[1] MongoDB documentation provides detailed guidelines on managing unstructured data and optimizing database performance for large-scale applications (MongoDB, 2023).
"Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks". Shaoqing Ren, Kaiming He, Ross B. Girshick, Jian.
[2] Express.js and Node.js offer comprehensive documentation and community resources that detail creating RESTful APIs and handling asynchronous operations, which are crucial for the efficient processing of image uploads and communication with Python services (Express.js, 2023; Node.js, 2023).
[3]React.js, known for its efficient rendering of interactive user interfaces, has extensive resources for building dynamic front-ends that interact seamlessly with back-end services (React.js, 2023).
[4] Vinyals, O., Toshev, A., Bengio, S., & Erhan, D. (2015). Show and tell: A neural image caption generator. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY RESEARCH

IN SCIENCE, ENGINEERING AND TECHNOLOGY

| Mobile No: +91-6381907438 | Whatsapp: +91-6381907438 | ijmrset@gmail.com |

www.ijmrset.com