Building the Image Module for the Creation Content Pipeline

Proposal Submitted by Group #14: Ishmaiah Etienne, Hunter Bresler and Alex Nikirk

27 September 2024

Problem Sponsor Name: Franklin Ludgood, Randall Denney

Organization Name: NETC - N5

Table of Contents

I. INTRODUCTION- Ishmaiah
..... 3

A. Background- Ishmaiah
..... 3

B. Statement of the Problem - Ishmaiah
..... 3

II. SCOPE OF WORK
..... 4

A. Overview- Ishmaiah
..... 4

B. Literature Review - Alex
..... 5

Image Processing
..... 5

Message Broker Systems
..... 5

C. Alternative Solutions - Alex
..... 5

Approach 1 - Centralized Image Processing Module
..... 6

Approach 2 - Distributed Microservices for Image Processing
..... 6

Approach 3 - Hybrid Design with Core Image Processing and Specialized Services .....
7

D. Evaluation – Alex
..... 7

E. Decision – Alex
..... 8

III. IMPLEMENTATION DETAILS
..... 8

A. System Specifications and Functionalities – Hunter
..... 8

B. Overall System Design with Block Diagram – Hunter
..... 8

IV.
References.....
..... 9

I. INTRODUCTION - Ishmaiah

This proposal responds to an RFP from the United States Navy, issued in [specific date if known, or title of RFP]. The United States Navy seeks a solution to streamline the organization, cataloging, and processing of its vast amount of training materials and manuals. The Navy requests the development of a series of modules that will be integrated with an existing message broker system to handle content such as documents and images.

The Navy's training materials are extensive and need to be efficiently managed to ensure timely access and optimized processing for future use. Last semester, a message broker system was developed as the communication tool of the content creation pipeline. The task now is to create specific modules that will attach to this system and handle document and image processing. Also, this module will be utilizing Docker for containerization and RabbitMQ for messaging between components.

By addressing this challenge, the proposed solution will allow for more efficient content management, accurate image and document processing, and seamless communication between all modules, contributing to the overall effectiveness of the Navy's training infrastructure.

A. Background - Ishmaiah

The United States Navy operates with a vast and complex set of training materials and technical manuals that are essential. These materials include documents and images that need to be managed effectively for efficient access, retrieval, and use.

To modernize and centralize the handling of these materials, the Navy initiated the Content Creation Pipeline project. Last semester, a group of students created a message broker system. This system allows for communication between the different modules. However, the current system lacks the necessary modules to process and store these documents and images in a structured, efficient manner.

B. Statement of the Problem - Ishmaiah

The core problem is the Navy’s need for an automated, scalable system that can efficiently process and manage its training documents and images. These materials are crucial for the work that they do.

The problem is technically challenging due to the following constraints:

1. **Machine Learning Integration:** Automated processing must leverage machine learning models, such as those built with PyTorch and TensorFlow, to classify images and extract meaningful data from documents.

2. **Inter-module Communication:** The solution must integrate with the previously developed message broker system (RabbitMQ) to facilitate seamless communication between various content processing modules.

To address this, the solution must meet specific performance criteria such as high availability and accuracy in image/document classification. A suitable solution must ensure the efficient management of training content, with the flexibility to scale for future needs.

II. SCOPE OF WORK

A. Overview - Ishmaiah

We propose to design and develop the Image and Document Module as a key component of the Navy's Content Creation Pipeline. This module will be integrated with the existing message broker system (RabbitMQ) and deployed in a cloud environment to ensure scalability. The work will be divided into several stages:

1 Research Stage

2 Conduct research on all the necessary tools needed to implement these modules.

1 Design Stage

2 Design the architecture of the Image Module, focusing on the integration with the message broker system.

3 Define the data processing pipeline for document and image tasks, including format validation, image classification, and metadata extraction.

1 Development Stage

2 Develop the module using machine learning models to automate the processing of documents and images.

3 Implement communication with RabbitMQ to allow for seamless exchange of messages between the modules and other components.

1 Testing and Calibration Stage

2 Test the system's performance in various scenarios, ensuring that it meets accuracy requirements for image classification.

3 Calibrate machine learning models to improve classification accuracy and optimize performance.

1 Deployment Stage

2 Ensure that the module integrates seamlessly with the rest of the content creation pipeline, providing real-time feedback and results.

By this project's end, the Navy will have a fully operational Image and Document Module, capable of processing and managing their training materials in a centralized, scalable, and automated manner. The module will provide critical functionality for the overall content creation pipeline, allowing for efficient document and image handling across the organization.

B. Literature Review - Alex

In this section we review existing research and technologies related to image processing, document management, and message broker integration. An understanding of this will help identify effective solutions.

Image Processing

The research paper *"Review of Deep Learning: Concepts, CNN Architectures, Challenges, Applications, Future Directions"* by Laith Alzubaidi et al. (2021) offers valuable insights into the evolution and application of deep learning models, especially Convolutional Neural Networks (CNNs). These insights are particularly relevant for developing the Navy's Image Module, as incorporating advanced CNN architectures will significantly enhance its capability to accurately classify and detect objects within training materials.

In the Navy's Image Module, we will utilize the pre-trained ResNet-50 model to ensure precise and efficient image categorization. By leveraging this pre-trained model, we can minimize the need for extensive datasets while still achieving high accuracy and eliminating the need to train our own classifier. Furthermore, the scalability and efficiency of ResNet-50 make it an ideal choice for deployment across various Navy operational environments.

Message Broker Systems

The integration of microservices architecture, supported by message broker systems like RabbitMQ, is crucial for enhancing the functionality of distributed systems. In their paper, *"A Systematic Mapping Study in Microservice Architecture,"* Alshuqayran et al. (2016) outline frameworks for effective microservices implementation, emphasizing key aspects such as scalability, fault tolerance, and asynchronous communication. RabbitMQ stands out as a robust solution for managing inter-service communication, ensuring reliable message delivery, and facilitating the seamless interaction of diverse modules.

By utilizing RabbitMQ, the Image Module will achieve efficient communication with other components of the Content Creation Pipeline. This integration will enhance the module's performance and scalability, allowing for the smooth processing of training materials and optimizing the overall content management system for the Navy's needs.

C. Alternative Solutions – Alex

In this section, we explore different design approaches considered for the Image Module. This module must efficiently process and manage a large volume of images, including classification, metadata extraction, and integration with the existing message broker system. Each alternative design is evaluated based on technical feasibility, performance, and integration capabilities.

Approach 1 - Centralized Image Processing Module

This approach focuses on developing a single, centralized Image Module that handles all aspects of image processing, including classification, resizing, metadata extraction, and storage management.

Strengths:

1. **Simplified Integration:** A single module simplifies integration with the message broker system and reduces the complexity of communication between different components.

2. **Unified Data Flow:** All image processing tasks are managed within one module, reducing the chances of data inconsistency and simplifying the debugging process.

3. **Lower Deployment Complexity:** Only one module needs to be deployed, making the deployment process straightforward and reducing overhead.

Weaknesses:

1. **Scalability Limitations:** The centralized nature of the module can create a bottleneck if the volume of images to be processed increases significantly.

2. **Performance Overload:** With all processing tasks in one module, there is a higher risk of overloading the system, which could lead to slower performance and delayed processing times.

3. **Maintenance Challenges:** Any changes or updates to the module require re-deployment of the entire module, potentially disrupting ongoing operations.

Approach 2 - Distributed Microservices for Image Processing

This approach involves breaking down the Image Module into smaller, specialized microservices. Each microservice is responsible for a specific task, such as image classification, metadata extraction, or storage management, and communicates with other services via the message broker system.

Strengths:

1. **Scalable Architecture:** Each microservice can be scaled independently based on the load, ensuring efficient resource utilization and higher performance.

2. **Modular Updates:** Individual services can be updated, tested, and deployed independently without impacting the overall system.

3. **Fault Tolerance:** Failure in one microservice does not affect the functioning of others, increasing the system's reliability and fault tolerance.

Weaknesses:

1. **Increased Complexity:** Managing multiple microservices requires sophisticated orchestration and monitoring tools, making the system more complex to develop and maintain.

2. **Communication Overhead:** Increased communication between microservices can lead to latency issues, especially when processing large volumes of images.

3. **Higher Deployment Overhead:** More components to deploy and monitor, increasing the operational overhead for the development team.

Approach 3 - Hybrid Design with Core Image Processing and Specialized Services

This hybrid approach combines a core Image Processing Module with specialized microservices for specific tasks like advanced image classification or metadata enrichment. The core module handles basic image processing and communication with the message broker, while specialized services manage resource-intensive or complex operations.

Strengths:

1. **Balanced Complexity:** By combining a centralized core with specialized services, this approach maintains manageability while allowing for scalable and flexible processing capabilities.

2. **Optimized Performance:** Critical tasks are handled by specialized services, which can be optimized for performance, while the core module ensures streamlined communication and basic processing.

3. **Simplified Maintenance:** The core module handles fundamental operations, reducing the need for frequent updates, while specialized services can be modified and deployed independently.

Weaknesses:

1. **Moderate Integration Effort:** Requires careful design to ensure seamless integration between the core module and specialized services, particularly for message handling and data consistency.

2. **Resource Management Complexity:** Balancing resource allocation between the core module and specialized services can be challenging, particularly under varying loads.

3. **Potential for Latency:** Communication between the core module and specialized services can introduce latency, particularly if not well-optimized.

D. Evaluation – Alex

1. **Technical Feasibility:** The ease with which each design can be implemented using existing technologies and skills.

1. **Performance:** The efficiency of image processing tasks, including speed and accuracy.

1. **Integration Capabilities:** The ability of the design to seamlessly connect with the existing message broker system and other components of the Content Creation Pipeline.

1. **Scalability:** The potential for the design to handle increased volumes of images over time without significant performance degradation.

1. **Maintenance and Upgradability:** The ease of updating or maintaining the system, including how changes to one component affect others.

1. **Operational Overhead**: The resources required for deployment and monitoring, including staff training and system management.

Each of these criteria is organized based on its importance from greatest importance to least importance. By prioritizing technical feasibility and performance, we aim to minimize the risks of errors and issues in this proof-of-concept project.

E. Decision – Alex

Given the evaluation criteria and the Navy's emphasis on simplicity for a proof of concept, we firmly believe that Approach 1, the Centralized Image Processing Module, is the most fitting solution for our immediate goals. This approach minimizes complexity while maximizing efficiency, allowing us to quickly demonstrate the module's capabilities. By consolidating all image processing tasks into a single module, we can ensure a more streamlined integration with the existing message broker system, reducing the potential for errors during deployment.

Furthermore, this centralized design allows for easier debugging and maintenance, which is crucial during the early stages of development. While scalability is a consideration, the initial focus is on proving the concept's viability; thus, a centralized architecture aligns perfectly with our current objectives. Ultimately, this approach not only satisfies the Navy's requirements for simplicity and reliability but also lays a solid foundation for future upgrades and changes.

III. IMPLEMENTATION DETAILS

A. System Specifications and Functionalities – Hunter

- The Image Module will be connected to a message broker system through RabbitMQ, which it uses to take and send requests between other modules in the overarching system.
- The Image Module will receive BSON files through the message broker system and parse them for necessary data and an image to classify.
- The Image Module will not run costly operations on duplicate image files. These image files will be checked for repeats using SHA-256 hashing.
- The Image Module will classify images it receives with metadata tags and keywords.
- The Image Module will send images and their associated metadata/keywords to the Metadata Module across the message broker system where the images will be packed with the metadata and stored in a database.

B. Overall System Design with Block Diagram – Hunter

The system is designed as a modular component that seamlessly integrates into the existing structure while enhancing its functionality. The Image Module serves as the computational hub, bridging user interactions from the dashboard module with the file management and storage capabilities of the metadata module. This module is specifically engineered to process images swiftly and accurately, operating with a layer of abstraction from the rest of the system.

IV. References

Alzubaidi, L., Zhang, J., Humaidi, A.J. *et al*. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *J Big Data* **8**, 53 (2021). https://doi.org/10.1186/s40537-021-00444-8

Alshuqayran, N., Ali, N., & Evans, R. (2016). A Systematic Mapping Study in Microservice Architecture. In 2016 IEEE 9th International Conference on Service-Oriented Computing and Applications (pp. 44-51). IEEE. https://doi.org/10.1109/SOCA.2016.15