DEVELOPMENT AND PRACTICAL APPLICATION OF AN ARTIFICIAL INTELLIGENCE-BASED IMAGE SEGMENTATION SYSTEM

Sadikov Rustam Tokhirovich,
Head of the Multimedia Technologies Department, Tashkent University of Information Technologies (TUIT).

Abidova Shaxnoza Bakhodirovna
Associate Professor, Tashkent University of Information Technologies (TUIT)

Samadov Dilshodbek Usmon o‘g‘li
Master's Student, Tashkent University of Information Technologies (TUIT)

Samadov Dilmurod Usmon o‘g‘li
Student, Tashkent University of Information Technologies (TUIT)

Abstract: This article discusses the development of a software system for automating the image segmentation process based on artificial intelligence. The software was developed using the Meta Segment Anything Model (SAM), enabling automatic object extraction from image content and visualization of segmentation results. The server side of the system was implemented using the FastAPI framework, while the artificial intelligence model was integrated through the PyTorch library. The application was developed in the PyCharm environment and performs functions such as processing user-uploaded images, generating segmentation outputs, and storing extracted objects. Experimental results demonstrated that the developed system provides high-accuracy segmentation performance across different image types. The proposed solution has practical significance in the fields of computer vision and intelligent information systems.

Keywords: artificial intelligence, image segmentation, Meta SAM, FastAPI, PyTorch, Computer Vision, object detection, software system.

Аннотация: В данной статье рассматриваются вопросы разработки программной системы для автоматизации процесса сегментации изображений на основе технологий искусственного интеллекта. Программное

обеспечение разработано на основе модели Meta Segment Anything Model (SAM), обеспечивающей автоматическое выделение объектов из изображения и визуализацию результатов сегментации. Серверная часть системы реализована с использованием фреймворка FastAPI, а интеграция модели искусственного интеллекта выполнена с помощью библиотеки PyTorch. Программа разработана в среде PyCharm и выполняет функции обработки загружаемых пользователем изображений, формирования результатов сегментации и сохранения выделенных объектов. Результаты экспериментов показали, что разработанная система обеспечивает высокую точность сегментации для различных типов изображений. Предложенное решение имеет практическое значение в области компьютерного зрения и интеллектуальных информационных систем.

Ключевые слова: искусственный интеллект, сегментация изображений, Meta SAM, FastAPI, PyTorch, компьютерное зрение, обнаружение объектов, программное обеспечение.

Introduction

With the rapid development of artificial intelligence technologies, the field of Computer Vision has become one of the most important areas of modern information technologies. In particular, automatic image analysis, object detection, and image segmentation tasks are widely applied in healthcare, industry, security systems, remote monitoring, and automated control systems. Image segmentation is the process of separating objects from the background within a digital image and determining the boundaries of each object, making it a fundamental component of computer vision systems.

Traditional image processing approaches often require manual parameter adjustment, complex configuration processes, and high computational costs. Furthermore, variations in image quality under different conditions may negatively affect segmentation accuracy. Therefore, segmentation methods based on Deep

Learning and artificial intelligence technologies have become a significant research direction in recent years[8].

Among recent advancements, the Segment Anything Model (SAM), developed by Meta, has gained considerable attention due to its capability to accurately identify and segment objects within images. The model significantly improves segmentation performance by enabling automatic object extraction across various image types.

In this study, an artificial intelligence-based system utilizing the Meta SAM model was developed for image segmentation tasks. The software solution was implemented using the FastAPI framework and the PyTorch library, providing capabilities for processing user-uploaded images, performing object segmentation, visualizing results, and generating extracted object outputs. The study further analyzes the system architecture, operational algorithm, utilized dataset, and experimental evaluation results[6].

Research Methodology

This study applied a methodology for developing and evaluating an artificial intelligence-based image segmentation system. During the research process, Computer Vision and Deep Learning technologies were employed to achieve automated object detection and segmentation within images. The Segment Anything Model (SAM), developed by Meta, was selected as the core model for performing object identification and segmentation tasks[2].

For software implementation, the FastAPI framework was utilized to develop the server-side architecture, while the PyTorch library was integrated for artificial intelligence model management and execution. The system was developed in the PyCharm programming environment, and data communication between the user interface and the server was established through REST API technology[7].

The proposed methodology consisted of several stages, including image acquisition, preprocessing operations, image segmentation using the SAM model, result generation, and visualization of segmented outputs. The developed approach enabled efficient image processing and automated object extraction, providing a reliable solution for computer vision applications.

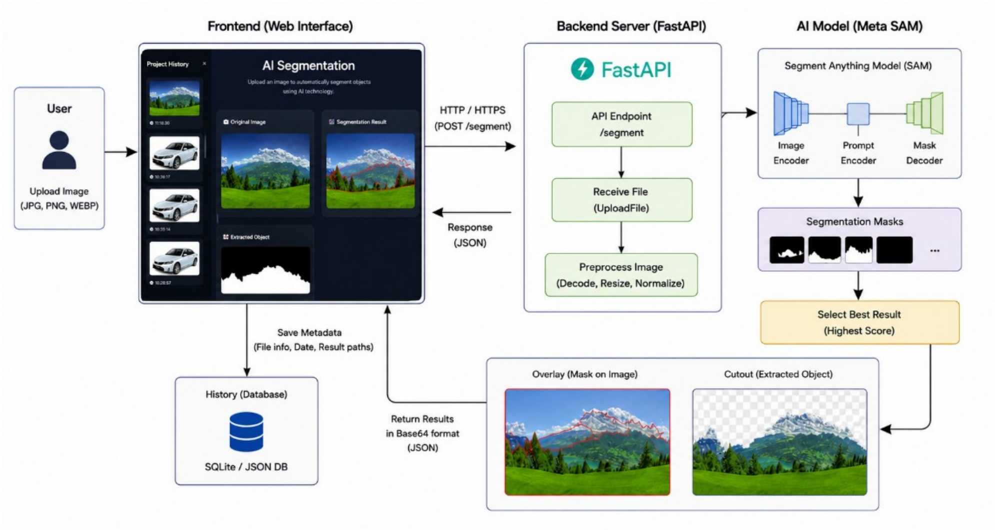

Figure 1. Architecture of the AI-Based Image Segmentation System

The research methodology consisted of several stages. Initially, an image acquisition and preprocessing mechanism was developed. Subsequently, the input images were transferred to the segmentation model, where generated segmentation masks were evaluated and ranked based on accuracy scores. In the final stage, the segmentation result with the highest confidence score was selected and presented to the user in a visual format[6].

To evaluate the effectiveness of the developed system, experimental tests were conducted using different categories of images. The segmentation accuracy and processing speed were analyzed to assess the performance of the proposed approach.

Dataset and Data Preparation Process

This study utilized various image datasets to develop and evaluate an artificial intelligence-based image segmentation system. The primary objective of the system was to automatically identify objects within images, separate them from the background, and generate segmentation outputs. Therefore, the dataset was constructed using real-world images containing diverse object categories.

The developed software supports two methods for data input. The first approach allows users to upload images directly through the graphical user interface. Users can provide images in JPG, PNG, and WEBP formats for segmentation processing. The second approach enables automatic image loading from existing files within the software environment. This strategy improves usability while optimizing the image processing workflow[1].

The dataset preparation process involved handling image data of different sizes, object categories, and visual characteristics to enhance the robustness of the segmentation model. Such diversity in the dataset contributes to improving model generalization capability and segmentation performance under different image conditions.

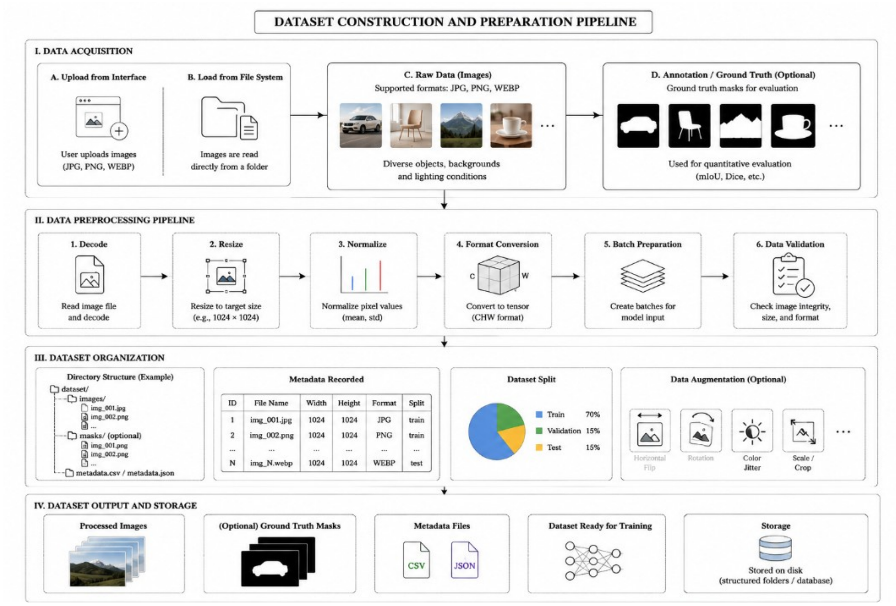

Figure 2. General architecture of dataset construction and data preparation process

The dataset included images containing various object categories, transportation vehicles, natural landscapes, daily-life objects, and background elements. The diversity of data characteristics improved the model's generalization capability and enabled segmentation performance evaluation under different conditions. To simulate real-world usage scenarios, images with different resolutions and quality levels were utilized[4].

Before being processed by the segmentation model, the images underwent several preprocessing stages. This phase plays a critical role in improving the efficiency and performance of the image segmentation system. Initially, uploaded images were processed through a decoding stage. Subsequently, image dimensions were adjusted according to model requirements through a resizing operation. In addition, a normalization procedure was applied to improve image processing quality and computational efficiency. These preprocessing operations ensured that all input data were transformed into a standardized format suitable for the segmentation model[3].

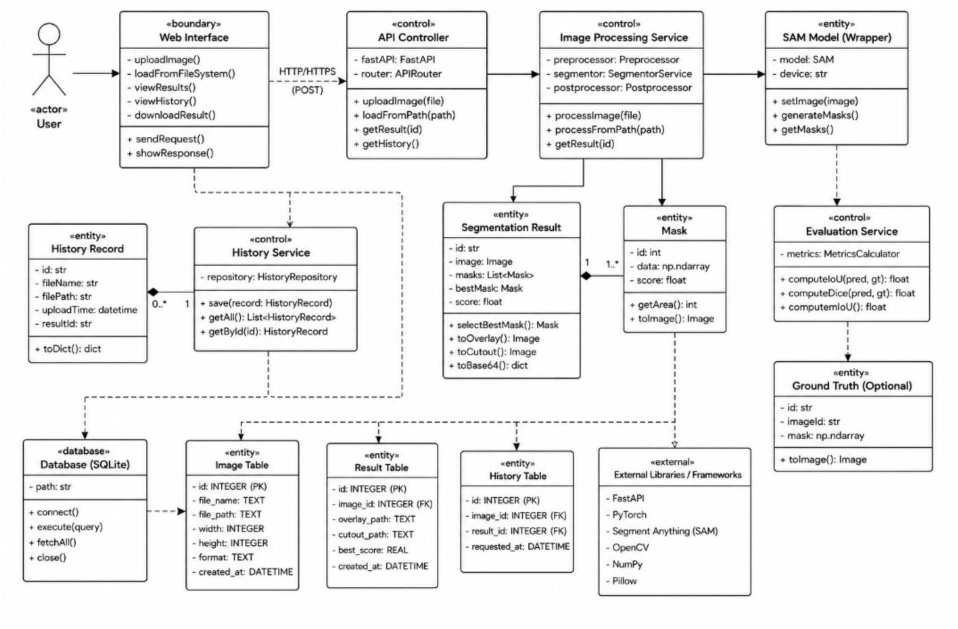

Figure 3. UML diagram of the AI image segmentation system and the relationships among its core modules

In the developed system, images are transmitted to the FastAPI server module and subsequently forwarded to the artificial intelligence model for processing. Object segmentation is performed using the Meta Segment Anything Model (SAM), where the segmentation mask with the highest confidence score is selected from the generated outputs. Based on the segmentation results, both an overlay image and a cutout image are generated.

The system also provides metadata storage functionality for uploaded images. Information such as image name, upload timestamp, and processing results is stored in the history module. This mechanism enables result tracking, data management, and further analytical processes[8].

The generated outputs are converted into Base64 format and transferred to the user interface. This approach improves cross-platform compatibility while simplifying data exchange procedures. Consequently, the proposed dataset preparation and preprocessing methodology contributes to enhancing artificial intelligence model performance and segmentation efficiency.

Proposed Algorithm and System Architecture

This study proposes a software system designed to automate image segmentation tasks using artificial intelligence technologies. The developed system was implemented to process uploaded images, automatically detect objects, and generate segmentation outputs. The system architecture consists of four major components: the user interface, server-side infrastructure, artificial intelligence model, and result management module[10].

In the first stage of the proposed algorithm, users upload images in JPG, PNG, or WEBP formats. The uploaded data are transferred to the FastAPI server and undergo preprocessing operations, including image decoding, resizing, and normalization.

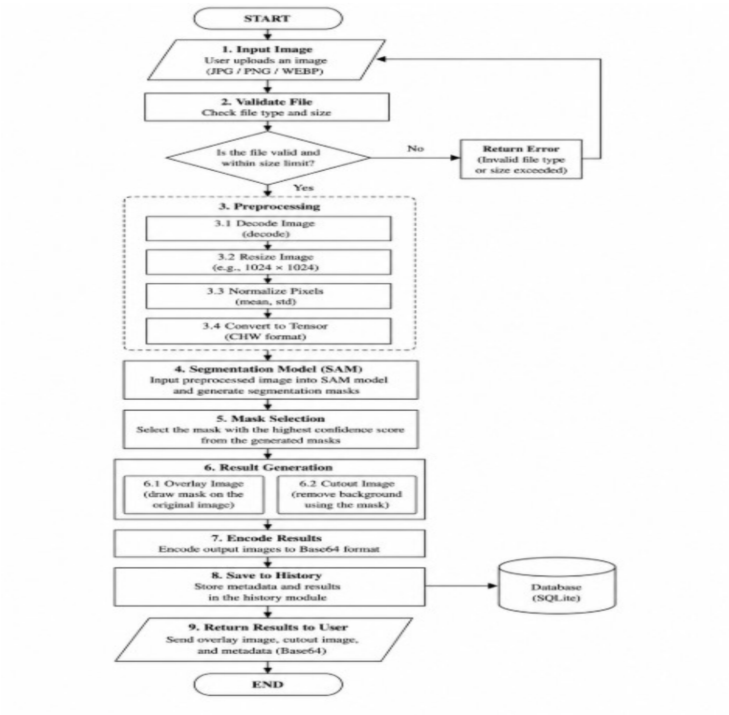

Figure 4. Proposed image segmentation algorithm illustrating the stages of image acquisition, preprocessing, segmentation using the SAM model, and final result generation.

The proposed algorithm follows a sequential processing pipeline designed for automated image segmentation tasks. Initially, input images are received from the user and transferred to the preprocessing module, where operations such as decoding, resizing, and normalization are performed. The processed images are then forwarded to the Segment Anything Model (SAM) for object detection and segmentation.

During the segmentation stage, the model generates multiple segmentation masks and evaluates them according to confidence scores. The mask with the highest confidence value is selected as the final segmentation output. The generated results include both overlay visualization and extracted object representation. This algorithmic structure improves segmentation efficiency, processing performance, and output accuracy for computer vision applications.

After preprocessing, the prepared input is forwarded to the Meta Segment Anything Model (SAM). The model analyzes image content and generates segmentation masks by identifying object boundaries. Finally, the segmentation output with the highest confidence score is selected and presented as the final result.

The segmentation results generated by the system are presented in two different forms. In the first approach, an overlay image is created by placing object boundaries directly onto the original image. In the second approach, a cutout image is generated by removing the background and extracting only the segmented object. The final outputs are converted into Base64 format and transmitted to the user interface, while processed results are stored in the history module for future access and reuse.

The proposed architecture was developed through the integration of FastAPI server technology, the PyTorch library, and the Meta Segment Anything Model (SAM). This architecture provides high processing speed, modularity, and compatibility with different image formats. As a result, the developed algorithm automates the image segmentation process and contributes to improving the efficiency of computer vision systems.

Software Implementation (FastAPI, PyTorch, SAM)

The artificial intelligence-based image segmentation system developed in this research was implemented using modern software technologies and deep learning approaches. The FastAPI framework was utilized for server-side development, PyTorch was employed for artificial intelligence model execution and computational operations, and the Segment Anything Model (SAM) was integrated to perform object segmentation tasks.

The backend infrastructure was implemented using the FastAPI framework, which enables REST API development, HTTP request processing, and efficient communication between the user interface and the server environment. FastAPI provides high-performance asynchronous processing capabilities, allowing the system to efficiently handle large-scale image processing operations and user requests.

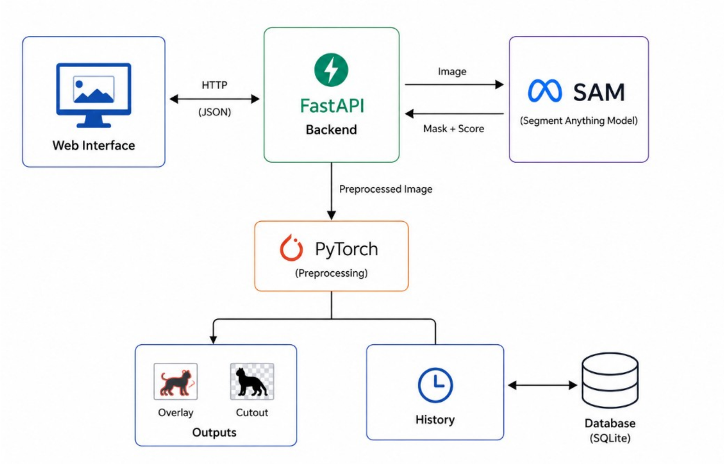

Figure 5. Software architecture of the developed image segmentation system based on FastAPI, PyTorch, and SAM technologies

PyTorch was utilized for image processing and artificial intelligence model management within the developed system. This library provides functionalities for neural network implementation, model loading, and tensor-based image processing operations. Uploaded images undergo several preprocessing stages, including decoding, resizing, and normalization before being transferred to the artificial intelligence model.

To perform object segmentation tasks, the Segment Anything Model (SAM), developed by Meta, was integrated into the system. The model identifies objects within image content and generates segmentation masks representing object boundaries. Among the generated masks, the output with the highest confidence score is selected as the final segmentation result.

The system produces outputs in two formats. The first output is an overlay image containing object boundaries superimposed on the original image, while the second output is a cutout image where background regions are removed to isolate the detected object. Final results are encoded into Base64 format and delivered to the user interface, while processed information is stored within the history module for future access.

The software was developed in the PyCharm programming environment and implemented using a modular architecture approach, allowing future expansion and integration of additional functionalities. This design improves system scalability and practical applicability.

Experimental Results and Analysis

Experimental evaluations were conducted to assess the effectiveness of the developed artificial intelligence-based image segmentation system. The experiments utilized JPG, PNG, and WEBP image formats to analyze segmentation accuracy and processing performance.

The obtained results demonstrated that the Meta Segment Anything Model (SAM)-based system effectively separates objects from backgrounds with high segmentation accuracy. Furthermore, FastAPI and PyTorch technologies enabled efficient image processing and rapid generation of segmentation outputs.

Experimental findings indicate that the developed software solution operates effectively across different image categories and possesses strong practical applicability in computer vision and image processing domains. Experimental findings indicate that the developed software solution operates effectively across different image categories and possesses strong practical applicability in computer vision and image processing domains. The proposed system provides reliable segmentation performance while maintaining processing efficiency for various image formats. The integration of FastAPI, PyTorch, and the Segment Anything Model contributes to improving both segmentation accuracy and system scalability. Furthermore, the modular architecture of the software enables future enhancements and the integration of advanced artificial intelligence technologies. These capabilities demonstrate the potential of the developed system for broader applications in intelligent visual analysis and automated image processing environments[7].

Table 1.

Analytical evaluation of the AI image segmentation system based on FastAPI, PyTorch and SAM model.

Test Image Type	Segmentation Quality	Processing Speed	Output Result	Analysis

Landscape Images	High	Fast	Accurate object boundaries	Stable segmentation performance in natural scene images.
Vehicle Images	High	Fast	Clear object extraction	High-quality mask generation and reliable cutout performance.
Complex Background Images	Medium-High	Moderate	Minor boundary variations	Segmentation remains effective under complex visual conditions.
PNG/ JPG/WEBP Formats	High	Fast	Consistent output	Supports multiple image formats without performance degradation.

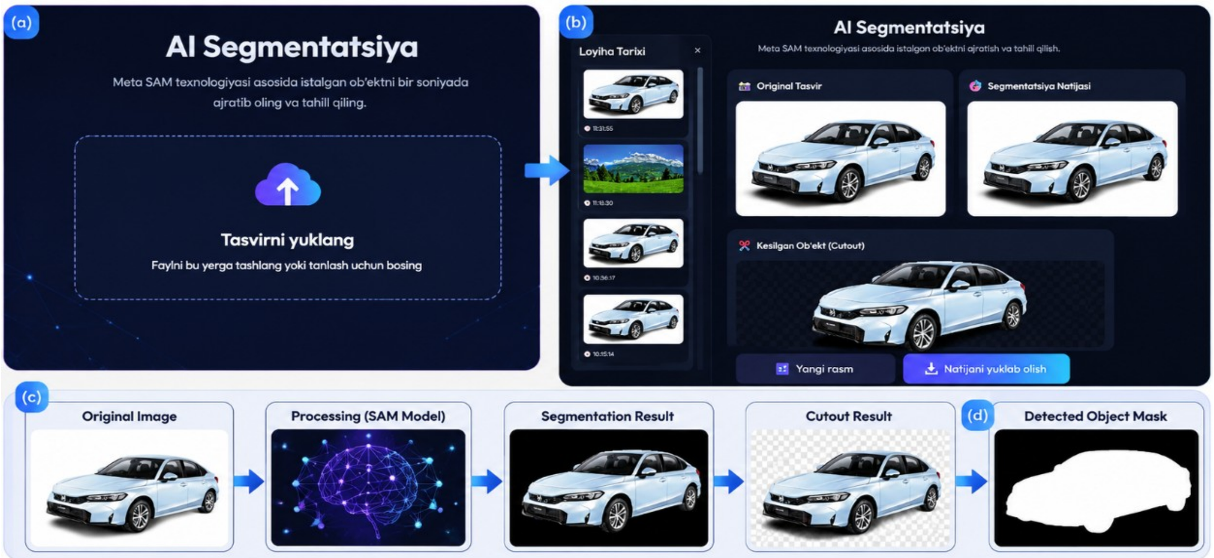

Figure 8. Experimental results of the developed AI image segmentation system demonstrating image upload, segmentation process, and cropped object extraction using the SAM model.

The figure presents the experimental results of the developed AI-based image segmentation system implemented using FastAPI, PyTorch, and the Segment Anything Model (SAM). The first stage demonstrates the image upload interface, where users provide input images for processing. The system then performs object segmentation by identifying and separating target objects from the background using the SAM model. The segmentation results are visualized through generated masks and processed output images. In addition, the cropped object extraction module produces isolated object representations for further analysis. The experimental results indicate that the proposed system provides accurate segmentation performance and efficient image processing capabilities. These outcomes demonstrate the applicability of the developed approach in computer vision and intelligent image analysis tasks.

Conclusion

This study presented the development and evaluation of an artificial intelligence-based image segmentation system. The software solution was implemented using the FastAPI framework, the PyTorch library, and the Meta Segment Anything Model (SAM) technology. The developed system provides functionalities for receiving user-uploaded images, performing preprocessing operations, segmenting objects, and visualizing segmentation results.

Experimental findings demonstrated that the proposed system can accurately detect and separate objects from backgrounds across different image formats. The FastAPI server architecture combined with PyTorch technology enabled efficient image processing and rapid result generation.

The obtained results indicate that the developed software has strong potential for practical applications in Computer Vision and image processing domains. Future work will focus on improving system performance, integrating additional artificial intelligence models, and extending real-time processing capabilities.

References

1. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W. Y., Dollár, P., & Girshick, R. (2023). *Segment Anything*. Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 4015–4026.
2. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., & others. (2019). *PyTorch:*

An Imperative Style, High-Performance Deep Learning Library. Advances in Neural Information Processing Systems, 32.

3. Chollet, F. (2018). *Deep Learning with Python*. Manning Publications.
4. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.
5. Howard, J., & Gugger, S. (2020). *Deep Learning for Coders with FastAI and PyTorch*. O'Reilly Media.
6. Grinberg, M. (2018). *Flask Web Development: Developing Web Applications with Python*. O'Reilly Media.
7. Ramírez, S., & Hernandez, J. (2022). *Computer Vision and Image Processing Techniques for Artificial Intelligence Systems*. Springer.
8. Szeliski, R. (2022). *Computer Vision: Algorithms and Applications*. Springer.
9. Bradski, G., & Kaehler, A. (2008). *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly Media.
10. Han, J., Pei, J., & Kamber, M. (2011). *Data Mining: Concepts and Techniques*. Morgan Kaufmann.