International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-13 Issue-4, April 2024

Dynamic Image Optimization and Code Generation Platform for Enhanced Data Augmentation

Saivijaay V K, S. Ganesh Kumar

***Abstract:** In the rapidly evolving domain of machine learning, the critical role of data quality, particularly image data, cannot be overstated. This research introduces a novel system uniquely designed to significantly improve the preprocessing and augmentation of image data for machine learning applications. At its core, the platform emerges as a comprehensive solution, meticulously bridging the gap between the acquisition of raw image data and its transformation into an optimized form ready for machine learning algorithms. What has been discovered, is a multifaceted system that not only simplifies the enhancement of image data but also elevates the quality of machine learning models by providing access to advanced image optimization techniques. The system distinguishes itself through a highly intuitive user interface that guides users in selecting and applying various optimisation strategies. These strategies are meticulously designed to enhance image quality and diversity, which in turn, can significantly improve the performance of machine learning models trained with such data. The platform's backend, powered by Python and leveraging libraries such as OpenCV, Pillow, and scikit-image, combined with a responsive frontend, ensures a seamless user experience and high-quality image processing. The generation of Python code for each processed image is a distinctive feature that enhances the platform's educational value, enabling users to learn, customise, and integrate optimisation techniques into their workflows. Moreover, the inclusion of an API extends the platform's utility beyond its web interface, facilitating the automation of data-augmentation pipelines and integration with external applications. This platform not only meets the immediate needs of data scientists and machine learning practitioners for data preprocessing and augmentation, but also contributes significantly to the field by promoting the understanding and application of image optimisation techniques.*

***Keywords:** Optimization, Codeless framework, Image preprocessing, Automation, Data preprocessing*

I. INTRODUCTION

This paper introduced a dynamic image optimization and code generation platform,

Manuscript received on 26 March 2024 | Revised Manuscript received on 03 April 2024 | Manuscript Accepted on 15 April 2024 | Manuscript published on 30 April 2024.

*Correspondence Author(s)

Saivijaay V K*, Department of Data Science and Business Systems, School of Computing, College of Engineering and Technology, Faculty of Engineering and Technology, SRM Institute of Science and Technology, Chennai (Tamil Nadu), India. E-mail: saivijaayvk21@gmail.com, sv7500@srmist.edu.in, ORCID ID: 0009-0007-8747-0118

Dr. S. Ganesh Kumar, Department of Data Science and Business Systems, School of Computing, College of Engineering and Technology, Faculty of Engineering and Technology, SRM Institute of Science and Technology, Chennai (Tamil Nadu), India. E-mail: ganeshk1@srmist.edu.in

An advanced web-based solution designed to address the challenges of data augmentation and optimization in a user-friendly and efficient manner. The genesis of this paper lies in the recognition of two fundamental challenges in the field of data science and AI: the labour-intensive process of manually optimising images for data augmentation and the gap in accessible tools that provide both optimised images and the means to reproduce such optimisations programmatically. To overcome these challenges, the platform offers an integrated solution that enables users to upload pictures, select the desired optimisation techniques through a simple interface, and receive the optimised image along with the corresponding Python code for the applied optimisations.

The platform serves a dual purpose: it acts as a practical tool for immediate image optimisation needs and as an educational resource that demystifies the image preprocessing process by providing users with the code behind the optimisation. This approach not only facilitates the practical application of data augmentation techniques but also enhances the user's understanding and ability to apply such techniques in their papers. In essence, this paper not only fills a critical gap in the toolkit of those working with image data but also aligns with the broader goal of advancing the field of AI and ML by making sophisticated data augmentation techniques more accessible and understandable. Through its dual focus on functionality and education, the platform is poised to have a significant impact on the way image data are processed and utilized in AI and ML applications.

II. LITERATURE SURVEY

The domain of image preprocessing and data augmentation plays a crucial role in preparing datasets for machine learning (ML) and artificial intelligence (AI) applications. As such, understanding the capabilities and limitations of existing systems is crucial for advancing this field. This literature survey examines the various challenges associated with current tools and technologies, providing a backdrop against which the necessity and potential impact of a new dynamic image optimisation and code generation platform can be evaluated.

A. **Complexity of User Experience:** Significant literature highlights the complexity of use as a major barrier to the widespread adoption of image preprocessing and augmentation tools. The steep learning curve associated with these powerful tools often deters users who lack a background in programming or image processing, thereby narrowing their utility to a subset of potential users (M.J.Smith 2006) [1].

B. **Local Laplacian Filters:** Edge-aware image processing with a Laplacian pyramid for capturing the edges of the images in an optimized way, where we can grasp the edges of the image [2].

C. **Fast Image Dehazing Method Based on Linear Transformation:** The Fast Image Dehazing Method Based on Linear Transformation (FIDMLT) presents a novel and efficient approach to image dehazing utilising linear transformation. Unlike traditional methods that depend on complex models and high computational load, FIDMLT swiftly estimates and corrects haze by applying linear transformations based on the image's colour and intensity profiles [3].

D. **Performance and Processing Speed:** Issues related to the performance and processing speed of current image preprocessing tools are well documented. These limitations are particularly critical for time-sensitive projects or those requiring real-time processing, affecting project timelines and deliverables (Garcia et al., 2019) [4].

E. **DSLR: Deep Stacked Laplacian Restorer for Low-Light Image Enhancement. The Deep Stacked Laplacian Restorer (DSLR) is an advanced deep learning model specifically designed to enhance low-light images**. By employing stacked Laplacian pyramids, DSLR effectively amplifies visibility, detail, and color fidelity in dimly lit photographs, while simultaneously minimizing noise. This approach ensures high-quality image restoration, preserving the natural aesthetics of the scene [5].

F. **Absence of Code Generation:** A notable gap in the literature is the lack of integrated code generation in existing platforms. This omission complicates the process of learning, automation, and reproducibility. Studies by Zhang et al. (2021) [6] highlight how the absence of code output for UI-driven changes impedes workflow efficiency and scalability in image processing tasks.

G. **The Impact of Pre- and Post-Image Processing Techniques on Deep Learning Frameworks: A Comprehensive Review for Digital Pathology Image Analysis:** This review paper examines the importance of Image preprocessing on the pathology image over a tissue. Techniques such as those used in CNN, and classification prediction for further analysis and compared over Pre & Post-processing of the image data [7].

H. **Image Preprocessing Techniques in Skin Diseases Prediction Using Deep Learning:** This review examines various image preprocessing techniques crucial for enhancing the performance of deep learning models in predicting skin diseases. Techniques such as normalization, augmentation, and segmentation are discussed, highlighting their role in improving model accuracy by refining image quality and highlighting relevant features for effective skin condition diagnosis [8].

This literature survey illuminates the pressing need for a platform that addresses these documented challenges. By offering an intuitive, flexible, and fully integrated solution for image preprocessing and data augmentation, the proposed platform stands to significantly enhance the efficiency, accessibility, and effectiveness of preparing data for AI applications. Such an innovation would not only bridge the existing gaps identified in the literature but also propel the application of advanced image processing techniques and streamlined workflows in the field forward.

III. OBJECTIVES

The primary objective of this paper was to develop a dynamic image optimization and code generation platform that streamlines the process of image preprocessing and augmentation for machine learning and artificial intelligence applications. This platform aims to provide an intuitive user interface that enables users to easily upload pictures, select from a variety of optimisation techniques, and receive both the optimised image and the corresponding Python code. By doing so, it seeks to bridge the gap between theoretical knowledge and practical applications, enhance data quality and diversity, and support reproducibility and scalability in data preprocessing workflows. Furthermore, the paper aims to democratise access to advanced image-processing tools, making them more accessible to a broader audience and thereby fostering innovation and learning within the AI and ML communities.

IV. METHODOLOGY

For the proposed dynamic image optimization and code generation platform, the system can be designed to ensure scalability, maintainability, and ease of use. Here is an overview of the different proposed modules, along with a brief explanation of their algorithms:

A. User Interface Module

To provide an intuitive web-based interface where users can upload images, select optimization techniques, and view both the optimized image and generated Python code.

Algorithm:

1. Input: The user uploads an image.
2. Process: The UI presents a list of available optimisation techniques, including resizing, normalisation, and augmentation techniques (e.g., rotation and flipping). The user selects the desired options and submits a request.
3. Output: The system displays the optimized image and provides the corresponding Python code snippet for selected operations.

B. Image Processing Module

To apply selected optimization techniques to uploaded images and generate an optimized output.

Algorithm:

1. Input: An image file and user-selected optimization parameters.
2. Process: For each optimization technique, apply the corresponding image-processing operation using libraries such as OpenCV, Pillow, or scikit-image [2]. Operations may include resizing, colour adjustments, geometric transformations, and filtering.
3. Output: Optimized image.

Retrieval Number: 100.1/ijeat.D443213040424
DOI: 10.35940/ijeat.D4432.13040424
Journal Website: www.ijeat.org

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

C. Code Generation Module

To generate readable and reusable Python code replicating the optimization steps applied to an image.

Algorithm:

1. Input: User-selected optimization parameters.
2. Process: Based on the selected operations, dynamically construct Python code snippets utilizing appropriate functions from image-processing libraries. Ensure that the code is well-structured and commented for optimal clarity.
3. Output: Python code that performs the selected image optimization operations when executed.

D. Integration and API Module

To enable easy integration with other applications and services, thereby facilitating automated workflows and broader applicability.

Algorithm:

1. Input: API requests with image data and optimization parameters.
2. Process: Validating and sparse API requests. Selected image optimizations were applied as per the Image Processing Module. Generate the corresponding Python code per the Code Generation Module.
3. Output: API responses with optimized images and Python code snippets.

E. Education and Resource Module

To provide users with resources, tutorials, and examples to understand and learn more about image-processing techniques and their applications in ML/AI.

Algorithm:

1. Input: User queries or selections of learning resources.
2. Process: Present curated content, tutorials, and code examples related to the selected image processing techniques. Provides interactive examples that enable users to observe the effects of various parameters and methods in real-time.
3. Output: Educational content and interactive learning tools.

Each of these modules works in concert to provide a comprehensive solution that not only streamlines the image optimisation process for AI/ML projects but also educates users on the underlying techniques, thereby enhancing their skill sets and understanding of image processing within the context of AI/ML applications.

V. ARCHITECTURE

1. **User Input Image:** The process begins with the user uploading an image to the platform for processing.

2. **Image Processing Steps:**

A. **Normalise:** The image undergoes normalisation, which adjusts the pixel values to a standard range, preparing it for further processing.

B. **Remove Noise:** Noise reduction algorithms are applied to the image to remove artefacts and enhance image quality.

C. **Image Enhancement:** Additional enhancement techniques are applied to improve the image's visual appeal, such as contrast adjustment or sharpening.

3. **Code Generation:** In addition to image processing, the platform generates the corresponding Python code, allowing users to understand the transformations and reproduce them later.

4. **Exact Output:** The system produces the optimized image as a direct output from the applied processing steps.

5. **User End Validation:** The user reviews the optimised image for quality and accuracy, ensuring it meets the specified requirements.

6. **Final Steps:**

A. Download Enabled: The platform enables users to download the optimised image.

B. Optimise: If the user is not satisfied with the result, there may be an option to optimise further or fine-tune the enhancements.

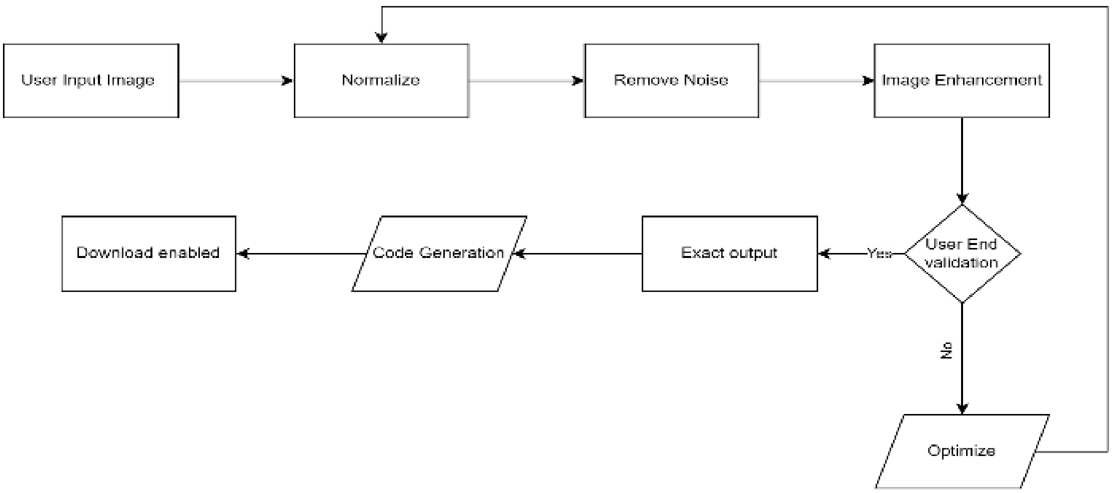

Fig. 5.1

This flowchart outlines a user journey through an image optimisation platform, highlighting both the image processing and educational components of code generation and user validation.

Novelty: - https://pypi.org/project/basic_image_preprocessing/

Dataset link: - https://www.kaggle.com/datasets/navoneel/brain-mri-images-for-brain-tumor-detection

VI. RESULT

A. Image Colour Type Selection

A. **RGB:** Ideal for images that will be displayed on screens. Our system supports complete RGB colour processing to maintain vibrant, true-to-life colours in your pictures.

B. Image Input Method

A. **File Upload:** Easily upload your image files using our secure and fast upload feature. We support a variety of file formats, including PNG, JPEG, and JPG, to ensure compatibility with most image types.

C. File Input Instructions

A. **Drag and Drop:** Conveniently drag and drop your image file into the designated area.
B. **File Size:** We accommodate large files up to 200MB, ensuring that even high-resolution images can be processed without compromise.
C. **Supported Formats:** Our tool accepts PNG, JPEG, and JPG formats, offering flexibility and convenience for your image pre-processing needs.
D. **Browse Files:** Navigate your local directories to select and upload your image file seamlessly.

D. Methods

The methods and options used to switch between different types of transformations that are needed for optimising images for Machine Learning classification or analysis. The list of methods mentioned below, such as

A. Linear Transformation
B. Non-Linear Transformation
C. Mathematical Transformation
D. Histogram Equalization
E. CLAHE Transformation
F. Laplacian Transformation
G. Canny Edge Detection
H. Edge Detection - User custom Matrix approach
I. Noise Filtration
J. Bilinear Filtration

E. Code Generation

Ultimately, if the user is satisfied with the preprocessing done by the tool, they can export the Python IPython notebook to integrate it into their project environment.

Figure ***6.1*** represents the whole user interface of the tool

Figure ***6.2*** represents the options of whether the user wants to upload a Black and white image or a colour image.

A. If the uploaded image is a colour image, the user can change the plane type from ***RGB*** to ***HSV*** to ***LAB***

Figure ***6.3*** represents the modules and the accessibility of the tool, and provides the code as well.

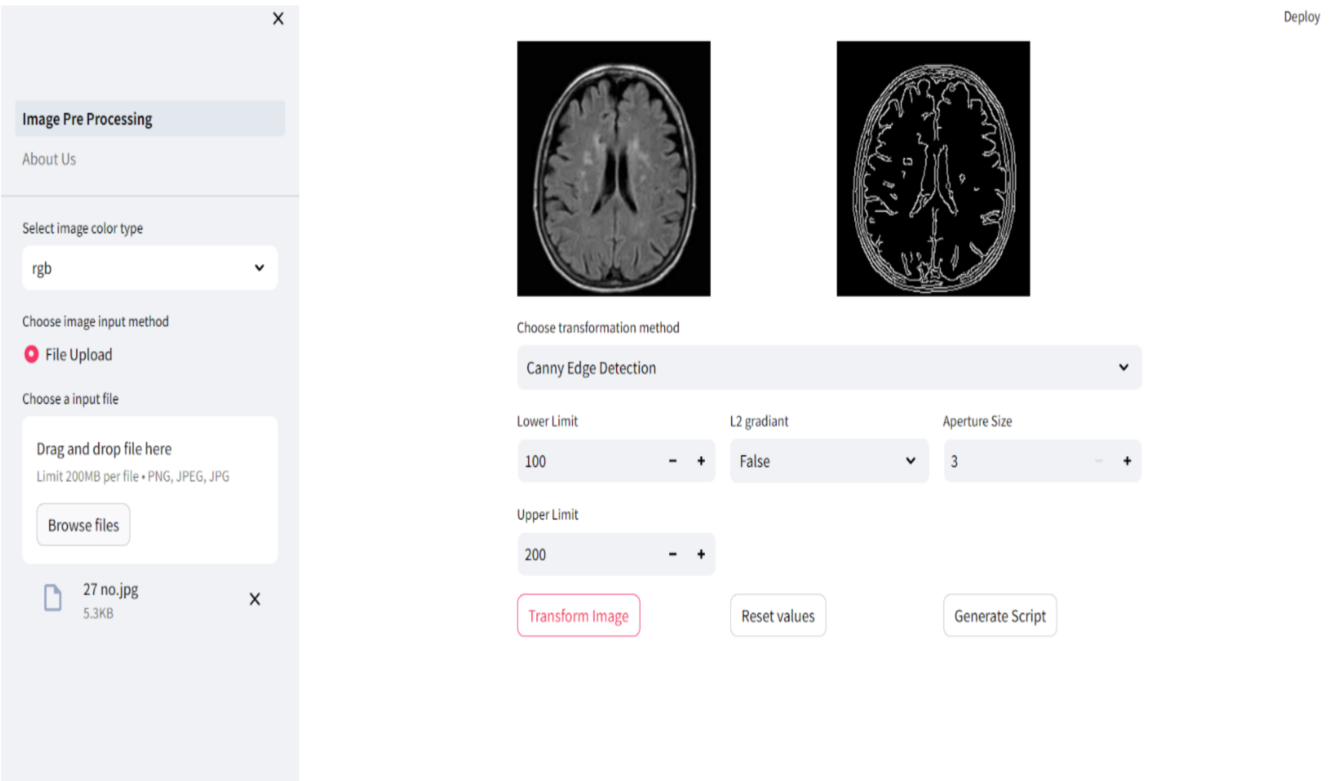

Fig. 6.1 Tool interface with Code Generation Enablement

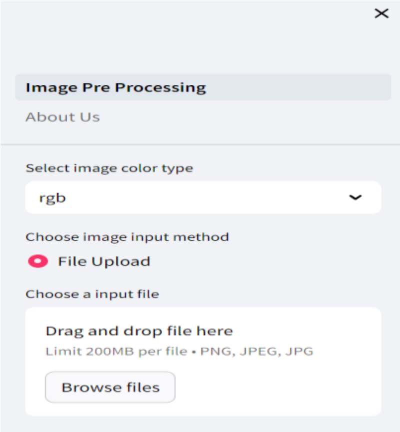

Fig. 6.2 User Interface Options

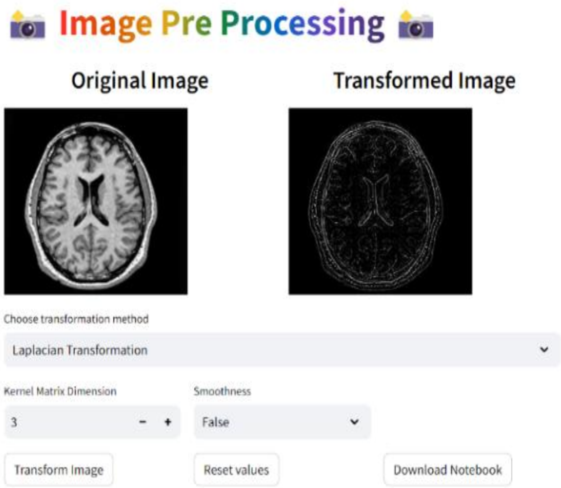

Fig. 6.3 User Interface Image Processing Methods

VII. CONCLUSION

Conclusively, the proposed Dynamic Image Optimization and Code Generation Platform marks a significant innovation in data augmentation and machine learning model training. This platform, by harnessing the power of advanced algorithms for dynamic image manipulation and automated code generation, not only revolutionizes the traditional approaches to data augmentation but also significantly enhances the efficiency and effectiveness of preparing datasets for machine learning applications. Through its capacity to automatically adjust images and generate corresponding code, it ensures that datasets are not only more diverse and reflective of real-world variability but also optimally formatted and ready for use in training more robust and accurate models. Moreover, the potential of this platform extends beyond the immediate improvements in data augmentation processes. By facilitating a more streamlined and automated approach to preparing training data, it paves the way for innovations across a broad spectrum of machine learning and AI-driven technologies. The integration of cutting-edge technologies, such as Generative Adversarial Networks (GANs) and automated scripting languages, could further enhance its capabilities, enabling users to create highly customised and sophisticated datasets with minimal effort. Ultimately, this Dynamic Image Optimization and Code Generation Platform stands at the forefront of a shift towards more intelligent and efficient data preparation tools. By providing a solution that not only simplifies but also elevates the quality of data augmentation, it contributes to the acceleration of machine learning research and the development of AI applications. This platform embodies the convergence of innovation and practicality, setting a new standard for how we approach the crucial task of data augmentation in the AI era.

DECLARATION STATEMENT

Funding	No, I did not receive.
Conflicts of Interest	No conflicts of interest to the best of our knowledge.
Ethical Approval and Consent to Participate	No, the article does not require ethical approval or consent to participate, as it presents evidence.
Availability of Data and Material/ Data Access Statement	Yes, the dataset used in the article is from Kaggle. The MRI brain images are also available in the dataset link provided in the comment. link:- https://www.kaggle.com/datasets/navoneel/brain-mri-images-for-brain-tumor-detection
Authors Contributions	Dr. Ganesh Kumar provided the use case, discussed the industry's challenges, and outlined an approach, in addition to the fieldwork and research conducted by Saivijaay V K.

REFERENCES

1. M.J.Smith, “Complexity and Ease of Use: A Design Study”, 2006, DOI:10.4017/gt.2006.05.02.008.00 https://doi.org/10.4017/gt.2006.05.02.008.00
2. Sylvain Paris, Samuel W. Hasinoff, Jan Kautz, “Local Laplacian filters: edge-aware image processing with a Laplacian pyramid” 2015 IEEE, DOI: https://doi.org/10.1145/2723694
3. Wencheng Wang; Xiaohui Yuan; Xiaojin Wu; Yunlong Liu, “Fast Image Dehazing Method Based on Linear Transformation”, 2017 IEEE, DOI: https://doi.org/10.1109/TMM.2017.2652069
4. Dávila Guzmán, M.A., Nozal, R., Gran Tejero, R. et al., “Cooperative CPU, GPU, and FPGA heterogeneous execution with EngineCL”, 2019, DOI - https://doi.org/10.1007/s11227-019-02768-y
5. Seokjae Lim; Wonjun Kim, “DSLR: Deep Stacked Laplacian Restorer for Low-Light Image Enhancement”. 2020 IEEE, DOI: https://doi.org/10.1109/TMM.2020.3039361
6. Zhen Li, Ying Jiang, The Automatic Code Generation, 2021, DOI: 10.1016/j. Procs.2020.02.099
7. Massimo Salvi a, U. Rajendra Acharya b c d, Filippo Molinari a, Kristen M. Meiburger, The impact of pre- and post-image processing techniques on deep learning frameworks: A comprehensive review for digital pathology image analysis, 2021 Elsevier, https://doi.org/10.1016/j.compbiomed.2020.104129
8. Anu V Kottath, SV Shri Bharathi, “Image Preprocessing Techniques in Skin Diseases Prediction Using Deep Learning: A Review”. 2022 IEEE, DOI: https://doi.org/10.1109/ICIRCA54612.2022.9985547

AUTHORS PROFILE

Mr. V K Saivijaay (M. Tech Student) is currently pursuing his Master of Technology in the Department of Data Science at SRM Institute of Science and Technology, SRM University, Chengalpat, India. He received his Bachelor of Technology (B.Tech.) in Mechanical Engineering in 2021 from Prince Shri Venkateshwara Padmavathy Engineering College, affiliated with Anna University. His research interests include Computer Vision, Machine Learning, and Deep Learning, among others.

Dr. S. Ganesh Kumar is currently working as a Professor in the Department of Data Science and Business Systems at the School of Computing, College of Engineering and Technology, Faculty of Engineering and Technology, SRM Institute of Science and Technology, Chennai, India. He has 22 years of work experience in research and academia. He received his Ph.D. degree in Computer Science Engineering from SRM Institute of

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

Science and Technology, SRM University, Chengalpat, India.