International Journal of Computing and Artificial Intelligence

E-ISSN: 2707-658X
P-ISSN: 2707-6571
www.computersciencejournals.com/ijcai
IJCAI 2025; 6(1): 95-98
Received: 03-01-2025
Accepted: 06-02-2025

Sachin Choudhari
Associate Professor, Computer Science and Engineering Jhulelal Institute of Technology Nagpur, Maharashtra, India

Bhushan Rajani
Computer Science and Engineering Jhulelal Institute of Technology Nagpur, Maharashtra, India

Aditya Jethani
Department Computer Science and Engineering Jhulelal Institute of Technology Nagpur, Maharashtra, India

Kashish Dhongde
Department Computer Science and Engineering Jhulelal Institute of Technology Nagpur, Maharashtra, India

Kshitij Wanjari
Department Computer Science and Engineering Jhulelal Institute of Technology Nagpur, Maharashtra, India

Saraswati Dudani
Department Computer Science and Engineering Jhulelal Institute of Technology Nagpur, Maharashtra, India

Shrawani Petle
Department Computer Science and Engineering Jhulelal Institute of Technology Nagpur, Maharashtra, India

Corresponding Author:
Sachin Choudhari
Associate Professor, Computer Science and Engineering Jhulelal Institute of Technology Nagpur, Maharashtra, India

Hueify-enhancing black and white images

Sachin Choudhari, Bhushan Rajani, Aditya Jethani, Kashish Dhongde, Kshitij Wanjari, Saraswati Dudani and Shrawani Petle

DOI: https://www.doi.org/10.33545/27076571.2025.v6.i1b.137

Abstract
Image coloring is an important task in computer vision, which includes color restoration of gray images. The traditional approach must be manually intervened, so the process time is consumed and dependent on expert knowledge. Nevertheless, the achievements of the deep learning field are possible to automatically paint, greatly reducing human efforts and increasing accuracy. This project presents a web application based on a plisk that performs automatic images using a pre -trained OpenCV in -depth education model. This model works in the color space of the laboratory where L channel (gray input) is processed, and the missing color channel A and B (color components) are predicted using deep training of the family unexplored neural network (CNN). This system implements a web platform that allows users to load gray images, and then processed through coloring models to generate realistic color output. The application provides an unbreakable user experience so that the user can see and load the processed image. Productivity assessment using structural similarity index (SSIM) and peak signal/ noise attitude (PSNR) is visually consistent and confirmed the effect of the model in the creation of the exact color for the situation. This system has actual applications in various fields, including historical restoration of photography, digital content improvement, medical visualization and media processing. The result shows that coloring based on deep learning can produce high quality results at a minimum cost of calculation. In the future, the improvement includes the user model training of deep learning, the output speed optimization for real -time processing, and the GENES (General Networks) to improve the quality of the color image and the further improvement of realism.

Keywords: Black-and-White Image Colorization, Deep Learning, Convolutional Neural Networks (CNNs), OpenCV, Blob Preprocessing, Bilinear Interpolation, Post-Processing, Image Restoration, Computer Vision, Image Enhancement, Historical Photo Restoration, Pre-trained Models, Automated Colorization

Introduction

The fascinating field of computer vision known as "black and white image colorization" is concerned with turning grayscale photos into vibrant, lifelike images. Through this procedure, monochrome photos get vitality and life, which increases their attractiveness and relatability to contemporary audiences. It is frequently used to improve media content, restore old photographs, and allow for imaginative artistic reimaginings.

A number of sophisticated methods and instruments are used in the process. A key component is OpenCV, a well-known open-source computer vision library. OpenCV offers fundamental features such picture scaling, color space conversions, and interaction with deep learning models, as Bradski and Kaehler [1] have described. It is an essential tool for this project and makes the difficult image processing chores simpler.

Convolutional Neural Networks (CNNs), which are strong machine learning models made to identify patterns and features in photos, provide the foundation of the colorization process. CNNs use their acquired knowledge of objects and scenarios to assess the grayscale image and forecast the right colors for various places. CNNs are excellent at recognizing textures, forms, and contexts, which allows them to produce realistic and context-aware colorizations, claim Liu *et al.* [2].

Another crucial step is blob preprocessing. In order to prepare the grayscale image, it must be resized to the appropriate size, its pixel values must be normalized, and key areas must be identified. These procedures guarantee that the system concentrates on important regions for increased accuracy and make the image compatible with the CNN model. Blob detection assists in finding pertinent portions of an image for subsequent tasks like colorization, as Ghosh [5] demonstrates.

The CNN's output is frequently of lesser resolution after it has made color predictions. The color channels are upscaled to the original image size using bilinear interpolation in order to remedy this. Bilinear interpolation ensures that the final result appears smooth and visually cohesive by smoothing transitions between neighboring pixels, as explained by Zhou *et al*. [3].

The colored picture is then refined using post-processing techniques. By adjusting saturation, contrast, and brightness, these methods enhance the colors' vibrancy and naturalness. Post-processing guarantees that the finished product is attractive and prepared for usage. According to VanceAI [4], these adjustments are crucial for improving the overall caliber of colorized photos.

This project provides a reliable method for colorizing black and white images by integrating CNNs for color prediction, blob preprocessing for input preparation, bilinear interpolation for scaling, and post-processing for enhancement. It is useful for applications in digital art, media improvement, and historical preservation because of its efficient, scalable, and high-quality outcomes.

Literature Survey

1. Deep Learning-Based Image Colorization

With developments in artificial intelligence and deep learning, the field of image colorization has undergone tremendous change. Conventional colorization methods were subjective and labor-intensive since they required manual intervention. Convolutional Neural Networks (CNNs) trained on massive datasets are used in deep learning-based colorization models to anticipate the missing color components in grayscale photos (Zhang *et al*., 2016) [6]. Without human help, the system can deduce and apply realistic colors by using a neural network to interpret grayscale inputs. The implementation of automated colorization systems has become quicker and easier with the use of pre-trained models, like those found in OpenCV.

2. OpenCV and Pre-Trained Colorization Models

OpenCV was first presented by Bradski and Kaehler (2008) [12] as a potent tool for machine learning, computer vision, and image processing. Pre-trained colorization models from OpenCV's deep learning module can be used in practical applications without requiring a lot of training. These models, which were created using extensive datasets, estimate the missing A and B color channels from a grayscale input (L channel) using the LAB color space. Because these models are so effective, they are ideal for web-based applications, like the Flask-based system used in this project, which enables users to upload grayscale photographs and get colorized outputs instantly.

3. Real-Time Image Processing and Automation

Image processing activities are now much more automated and efficient thanks to deep learning frameworks. According to Simonyan and Zisserman (2015) [11], good color prediction depends on high-quality feature extraction, which is made possible by deep convolutional architectures like VGG networks. Similar to this, ResNet, which He *et al*. (2016) [9] presented, enhanced deep networks by introducing residual learning and minimizing problems such as vanishing gradients. The ability of contemporary picture colorization systems to produce realistic and eye-catching colorized images has been improved by the widespread adoption of these designs.

4. Generative Adversarial Networks (GANs) for Enhanced Colorization

Generative Adversarial Networks (GANs) are one of the most recent developments in automatic image colorization. In order to improve colorization quality, Iizuka *et al*. (2016) [8] suggested an end-to-end learning strategy that makes use of both global and local priors. According to Ulyanov *et al*. (2018) [5], GANs produce high-quality images by honing the expected color distribution through the training of a generator-discriminator model. These techniques produce more brilliant and natural hues than conventional CNN-based models, which makes GANs a promising avenue for further advancements in deep learning-based colorization.

5. Applications of Image Colorization in Real-World Scenario

Automatic image colorization is used for purposes other than photography and entertainment. According to Reinhard *et al*. (2001) [13], color transfer techniques have been extensively employed in film enhancement, medical imaging, and historical photo restoration. Additionally, data preservation, forensic investigation, and improving accessibility for colorblind people can all benefit from AI-powered colorization. Additionally, augmented reality (AR) applications have investigated real-time image colorization, where AI-generated colors might enhance user interaction with grayscale images in virtual design, teaching, and gaming.

Existing Work

The creation of deep learning-based automatic image colorization requires a number of essential elements. Backend processing, deep learning model integration, picture preparation, and an intuitive web interface are all part of the system architecture. After processing grayscale photos into LAB color space, a pre-trained OpenCV colorization model is used to anticipate the color components that are missing.

Python, OpenCV, and NumPy are used in the software implementation, while Flask is used for backend functions. Users can upload grayscale photographs and obtain colorized results via the frontend, which was created with HTML, CSS, and JavaScript. RESTful APIs are used by the system to perform API requests, guaranteeing smooth front-end and back-end communication.

Large-scale processing may be done through cloud integration, and uploaded and processed photos are kept in temporary storage for data management. Secure file handling and SSL/TLS encryption are two examples of security methods that guarantee safe operations. Additionally, the system is tested and validated through user testing for quality evaluation, integration testing for API interfaces, and unit testing for model accuracy.

Finally, the deployment step makes sure that the system has maintenance and update provisions and is optimized for real-world use. An effective, precise, and intuitive image colorization technique that may be used for creative, media enhancement, and historical restoration is ensured by this methodical approach.

Methodologies
Deep learning models, backend processing, and a web-based user interface are all integrated into the systematic execution of the automatic image colorization system.

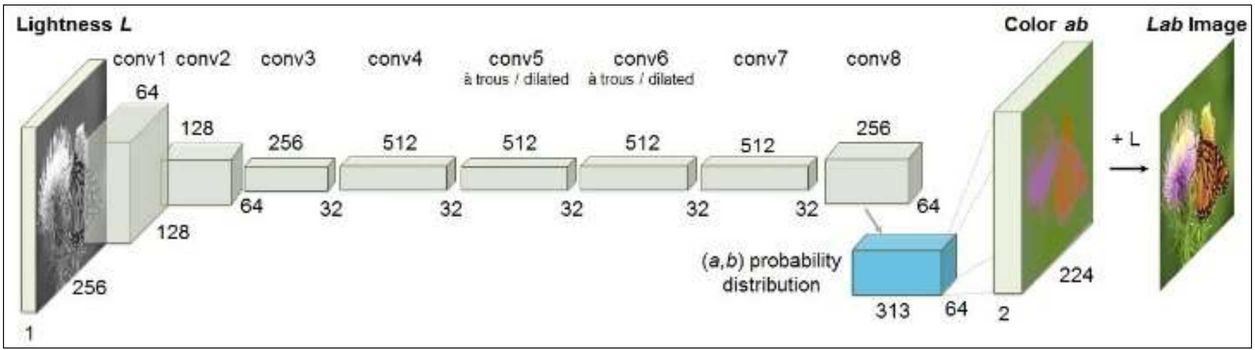
- **System Design and Architecture:** Specify the general structure, encompassing the preprocessing pipeline, backend/frontend integration, and deep learning model selection. Determine the required software (Python, OpenCV, Flask, TensorFlow/PyTorch) and hardware (GPU for acceleration, cloud storage).
- **Model Integration and Image Processing:** Apply colorization using OpenCV's pre-trained deep learning model. Utilize the deep learning model to forecast the absent A and B color channels after converting grayscale photos to LAB color space and extracting the L-channel.
- **Backend Development:** Use Flask as the server framework for processing, uploading images, and communicating via APIs. Images should be temporarily stored for processing and retrieval.
- **Frontend Development:** Create an intuitive web interface for image uploads, processing status, and result visualization using HTML, CSS, and JavaScript. For smooth user engagement, make sure the user interface is responsive and easy to use.
- **Database and Data Management:** If necessary, employ SQL (MySQL, PostgreSQL) or NoSQL, (MongoDB) databases to store user information and processed photos.

Testing and Optimization

- **Component Testing:** Verify API interactions, model effectiveness, and image processing accuracy.
- **Software Debugging:** Reduce latency and optimize the Flask backend for a seamless user experience.
- **User Testing:** Get opinions on how well the colorization is done and, if need, improve the model.
- **Performance Optimization:** Reduce computational overhead and speed up picture processing by increasing algorithm efficiency.

Maintenance and Deployment: Install the Flask-based program on a cloud platform or local server. Make plans for frequent updates, problem corrections, and maybe retraining the model to increase the accuracy of colorization.

Working



The suggested image colorization system must be implemented in a number of crucial steps. First, the necessary features and functionalities are outlined in the system design and architecture. A deep learning model predicts the missing A and B channels for realistic colorization, while the image processing phase starts with loading grayscale photos, transforming them into LAB color space, and extracting the L-channel.
The backend, which controls picture uploads and processing, is created during the software development phase using Flask, OpenCV, and Python. The colorization model processes the grayscale images and reconstructs them using expected color components. It is built on OpenCV's pre-trained deep learning network. For accurate representation, the output is subsequently transformed back to RGB color space.
The backend, which controls picture uploads and processing, is created during the software development phase using Flask, OpenCV, and Python. The colorization model processes the grayscale images and reconstructs them using expected color components. It is built on OpenCV's pre-trained deep learning network. For accurate representation, the output is subsequently transformed back to RGB color space.
HTML, CSS, and JavaScript are used to create the frontend development, which offers a straightforward and engaging experience for users to upload grayscale photographs and get colorized results. To ensure seamless user interaction, the processed photos are dynamically served by the Flask backend.
Flask APIs manage requests between the frontend and backend for system communication, guaranteeing effective picture transfer. Before being sent back to the user, the image processing pipeline makes sure that uploaded images are properly formatted and processed.
Testing and validation are performed on the system, including user testing to assess colorization quality, integration testing for Flask-OpenCV interfaces, and unit testing for model performance. During the deployment phase, the system is made completely functional and optimized for processing speed and image quality.
The system's efficiency, ease of use, and ability to generate high-quality colorized images are guaranteed by this methodical approach, which makes it valuable for AI-driven artistic applications, digital media enhancement, and historical restoration.

Conclusion
An important development in computer vision and digital restoration is the creation of the automatic picture colorization system. A Flask-based web interface and OpenCV's pre-trained deep learning model are combined to

provide an effective and user-friendly method for turning monochrome photographs into realistically colored ones. Because to this automation, colorization is no longer necessary, saving time and effort while maintaining high-quality output. Colorized photos may be uploaded and retrieved with ease using the straightforward online interface, which makes it suitable for AI-driven art, digital media enhancement, and historical restoration. This method can be used in more areas than just image enhancement, such as medical imaging, content production, and archival preservation. In the future, it might be possible to increase color accuracy by integrating Generative Adversarial Networks (GANs), optimizing real-time processing, and fine-tuning models using custom datasets. With its scalable and efficient solution for restoring and improving grayscale photos with little human involvement, this research showcases the potential of AI-driven automation in image processing.

References

1. VanceAI. Advanced Post-Processing Techniques for Image Enhancement [Internet]. 2023 Apr 5 [cited 2025 Apr 3]. Available from: https://vanceai.com
2. Ghosh P. The Role of Blob Detection in Preprocessing for Image Recognition Tasks [Internet]. 2022 Mar 10 [cited 2025 Apr 3]. Available from: https://www.researchgate.net/publication/386411243_Performance_Evaluation_of_Blob_Detection_Techniques_Using_Image_Processing
3. Zhou X, Sun Y, Wang L. Efficient Image Colorization Using Bilinear Interpolation Techniques. In: Conference on Image Processing Advances; 2020 Sep. p. 245-251. Available from: https://www.researchgate.net/publication/282449606_A_Review_Image_Interpolation_Techniques_for_Image_Scaling
4. Liu X, Li M, Yu T. Colorization of Grayscale Images Using Convolutional Neural Networks. J Image Process Anal. 2019 Oct;12(4):123-9. Available from: https://3dvar.com/Jwa2021Grayscale.pdf
5. Ulyanov D, Vedaldi A, Lempitsky V. Deep Image Prior. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2018 Jun. p. 9446-54. Available from: https://arxiv.org/pdf/1711.10925
6. Zhang R, Isola P, Efros AA. Colorful Image Colorization. In: European Conference on Computer Vision (ECCV); c2016 Oct. p. 649-66. Available from: https://arxiv.org/pdf/1603.08511
7. Larsson G, Maire M, Shakhnarovich G. Learning Representations for Automatic Colorization. In: European Conference on Computer Vision (ECCV); c2016 Oct. p. 577-93. Available from: https://arxiv.org/pdf/1603.06668
8. Iizuka S, Simo-Serra E, Ishikawa H. Let There Be Color!: Joint End-to-End Learning of Global and Local Image Priors for Automatic Image Colorization. ACM Trans Graph. 2016 Jul;35(4):1-11. Available from: https://arxiv.org/pdf/1603.08511
9. He K, Zhang X, Ren S, Sun J. Deep Residual Learning for Image Recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR); c2016 Jun. p. 770-8. Available from: https://arxiv.org/pdf/1409.1556
10. Gatys LA, Ecker AS, Bethge M. A Neural Algorithm of Artistic Style. J Vision. 2015 Nov;16(12):1-11. Available from: https://arxiv.org/pdf/1508.06576
11. Simonyan K, Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition. In: Proceedings of the International Conference on Learning Representations (ICLR); c2015 May. p. 1-14. Available from: https://arxiv.org/pdf/1409.1556
12. Bradski G, Kaehler A. Learning OpenCV: Computer Vision with the OpenCV Library. 1st ed. O'Reilly Media; c2008. p. 1-20. Available from: https://www.bogotobogo.com/cplusplus/files/OReilly%20Learning%20OpenCV.pdf
13. Reinhard E, Ashikhmin M, Gooch B, Shirley P. Color Transfer between Images. IEEE Comput Graph Appl. 2001 Sep/Oct;21(5):34-41. Available from: https://www.cs.tau.ac.il/~turkel/imagepapers/ColorTransfer.pdf