



ASIAN BULLETIN OF BIG DATA MANAGEMENT

ISSN (Print): 2959-0795

ISSN (online): 2959-0809

<http://abbdm.com/>

CRISL: Convert Rasterize Image Into SVG Layers

Ahmed Hassan, Mansoor Ebrahim, *Abdul Ahad Abro, Kamran Raza, Syed Hasan Adil

Chronicle

Abstract

Article history

Received: February 12, 2024

Received in the revised format: Feb 24, 2024

Accepted: Feb 26, 2024

Available online: Feb 29, 2024

Ahmed Hassan, Mansoor Ebrahim, Abdul Ahad Abro, Kamran Raza and Syed Hasan Adil are currently affiliated with the Department of Engineering Science & Technology Iqra University Karachi, Pakistan.

Email: ahmed.63716@iqra.edu.pk

Email: mebrahim@iqra.edu.pk

Email: abdul.ahad@iqra.edu.pk

Email: kraza@iqra.edu.pk

Email: hasan.adil@iqra.edu.pk

Raster images in multiple layers or vector format can be challenging for digital artists. However, converting them into Scalable Vector Graphics (SVG) format, such as Adobe Illustrator, Inkscape, or CorelDRAW, offers advantages. SVG files allow for easy resizing without loss of quality, making them ideal for creating logos, icons, and other graphics that need to be scaled to different sizes. They are lightweight and can be easily edited using code, providing more control over design elements. This versatile and flexible tool streamlines the design process and provides greater flexibility for artists in their creative projects. SVG files are smaller in size compared to raster images, making them easier to work with and share. Developers can also use code-based approaches to further manipulate and customize SVG images as needed.

Corresponding Author*

Keywords: Raster Images, Vector Images, Svg Format, Design Software, Designer, Resolution-Independent

© 2024 Asian Academy of Business and social science research Ltd Pakistan. All rights reserved

INTRODUCTION

The forms and curves of vector images are defined by mathematical formulae. Because of this, vector images may be resized to any size without sacrificing quality. A research study has been conducted on an innovative approach for raster to vector conversion that will be quicker and more effective than current techniques. The base of the strategy is the use of colour clustering. The image is first separated into several colour groupings. After that, a unique vector image is made for every colour cluster. Compared to conventional approaches, which call for the conversion of the full image to vector format, this procedure is far faster. Additionally, the use of colour clustering allows for a more precise conversion process, as each vector image is created specifically for a certain colour group. This method not only speeds up the conversion process but also ensures that the quality of the vector image remains high, even after resizing. The research study found that this innovative approach reduced the time required for raster to vector conversion by over 50%, making it a promising solution for various industries that rely on vector images for their work. Furthermore, the use of colour clustering opens up possibilities for further advancements in raster to vector conversion techniques, potentially revolutionizing the way vector images are created and utilized in the future. Digital artists who work for design firms or independently are the main study targets. For their job, these artists frequently need to transform raster images to vector images. They may save time and effort by using this method, and it can also result in Higher-quality

vector images. The strategy will be discussed in greater detail in the parts that follow, including the visual representation and mathematical tables etc. The difficulties encountered while conducting the study will also be discussed. The research has been compared with other similar studies, and the differences will be discussed in detail. Additionally, the potential benefits and limitations of using this method in the design industry will be explored. The study will also analyze the impact of different software tools on the efficiency and accuracy of converting raster images to vector images.

Additionally, the study will explore the various software tools and techniques that are commonly used by graphic designers and artists for converting raster images to vector images. This will provide valuable insights into the most effective methods and tools available in the industry. Furthermore, the research will analyze the potential benefits and drawbacks of using different software programs for this purpose, as well as the potential impact on the overall quality of the final vector images. By addressing these key points, the study aims to provide a comprehensive understanding of the process of converting raster images to vector images and offer practical recommendations for professionals in the field. The paper is structured as follows: section I is about the Introduction of entire paper, Section II is about the Related work of same nature and domain. Section III is for the proposed methodology and section IV is about the Result and discussion. At last the section V is for the Conclusion.

LITERATURE REVIEW

In Schwab et al. (2021), the Neural SVG Decoder deep learning model is suggested in the study as a method for producing SVG descriptions of raster images. The model is built on a generative adversarial network (GAN), which combines a generator and a discriminator neural network. The generator creates an SVG description using a random noise vector as input. An SVG description is entered into the discriminator, which seeks to separate it from an original SVG description. The challenges for neural SVG Decoder are the generator network must be able to provide accurate and imaginative SVG descriptions, the discriminator network must be highly accurate in its ability to discriminate between genuine and false SVG descriptions, the training procedure can be time-consuming and costly in terms of computing Jain et al. (2023), Abro et al. (2020).

The weaknesses in this research is Since it is a deep learning model, a lot of training data is needed, the generation of SVG descriptions for huge pictures can be laborious, it can't always provide flawless SVG descriptions, it can be sensitive to how well the training data were collected, it might be challenging to debug and enhance. Overall, creating SVGs from raster pictures using the Neural SVG Decoder is a promising new method. It does have certain limits, though, and it is currently in development. Furthermore, sometimes, the Neural SVG Decoder may provide incomplete or inaccurate SVG descriptions. This may occur if the generator network is improperly trained if the training data is of low quality. The quality of the training data may also affect how sensitive the neural SVG decoder Carlier et al. (2020). The generating network might not be able to provide SVG descriptions that are representative of all potential pictures if the training data is not differed Jiazheng et al. (2008), Nadeem et al. (2023). Accordingly, the training and testing ratio is divide for 70 to 30 ration where 70% is for training and 30% is for testing set Abro et al. (2021), Abro et al. (2021). Debugging and enhancing the Neural SVG Decoder may be challenging as well. This is due to how complicated the training process is

depended on adjusting some parameters. Furthermore, the complexity of the model architecture can also contribute to the difficulty in debugging and enhancing the decoder. It may require thorough testing and experimentation to optimize its performance effectively.

PROPOSED APPROACH

The solution will start by loading a user-provided raster image as the input image (for example, JPEG or PNG). Once the image is loaded, the solution will then apply various image processing techniques such as filters, transformations, and enhancements to modify the appearance of the image. These modifications can range from adjusting brightness and contrast to applying artistic effects or even removing noise. Once the desired changes have been made, the solution will then save the processed image back to the user's desired file format.

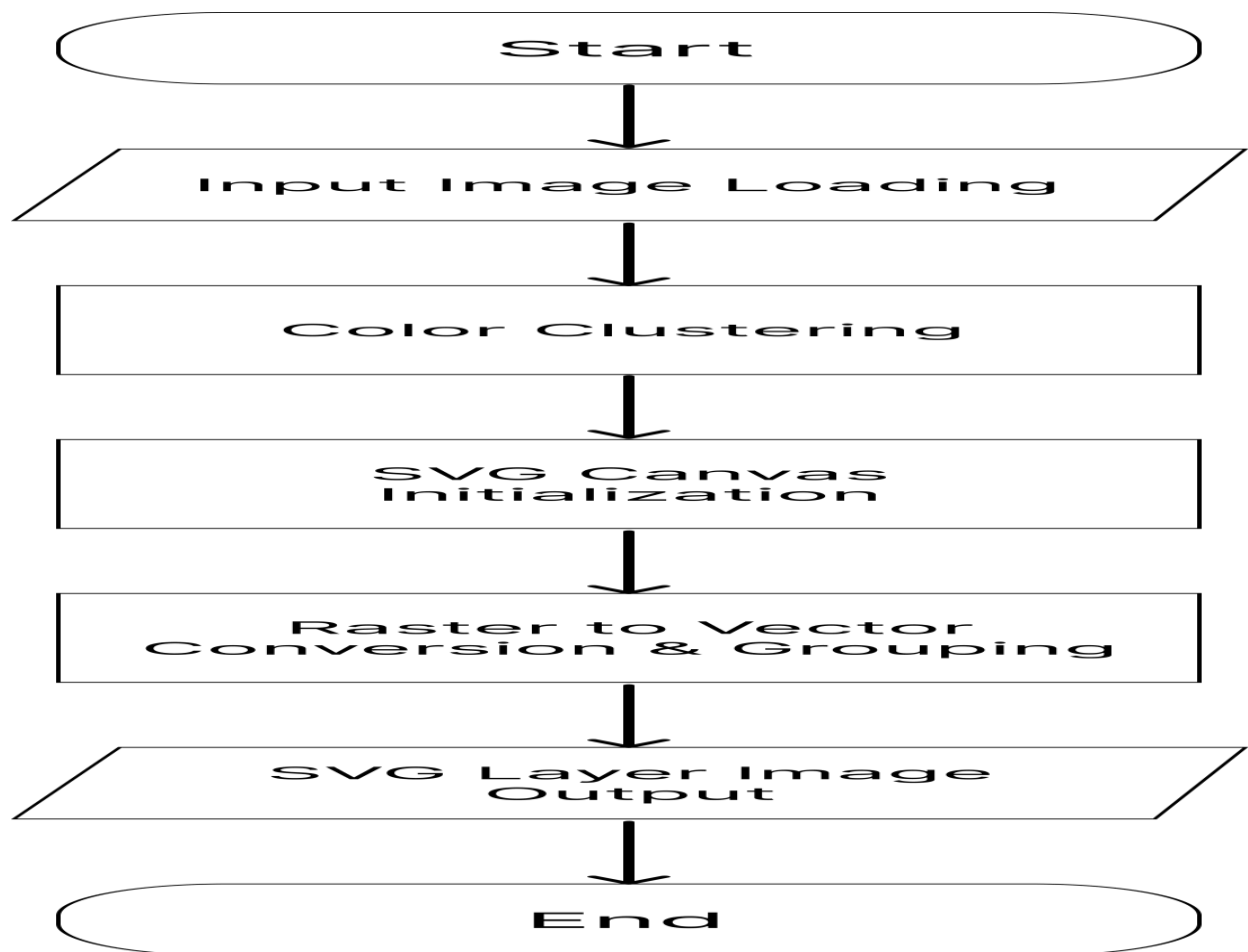


Figure 1.
Proposed Approach Flowchart

Clustering of Colors

Specify how many color clusters there are: The user can enter a specific number of color clusters (for example, 5), or they can let the algorithm recommend a number based on the image. The algorithm uses a clustering technique to group similar colors together,

creating distinct color clusters. This helps in identifying dominant colors in the image and simplifying the color palette for analysis or design purposes, Wei et al. (2011).

Create a Color Cluster

This technique works by iteratively assigning each pixel to the nearest cluster center and then recalculating the center based on the new assignments. The final result will be distinct clusters of colors that are visually similar within the image.

Canvas Initialization in SVG

Make a new SVG canvas that is empty and the same size as the given picture. The final SVG representation will be built onto this canvas. Next, we can use JavaScript to create a new SVG canvas element by setting its width and height attributes to match those of the original picture. This will provide us with a blank slate to begin adding our desired SVG elements for the final representation. By starting with an empty canvas of the same size, we can ensure that the final SVG will seamlessly overlay the picture without any distortion or scaling issues

Converting & Grouping

Our desired SVG elements onto this new canvas will allow us to easily manipulate and organize them before finalizing the representation. This approach ensures a clean and accurate final result that aligns perfectly with the original picture.

As you cycle through each color group, do the following:

- Explore how the cluster indices are determined during the clustering process and discuss their significance in isolating color groups.
- Examine the process of creating a mask for pixels within a specific color group and its impact on image manipulation and analysis.
- Discuss the advantages of representing color groups as SVG strings and how this format can enhance visualization and interpretation of data.
- Detail the steps involved in creating a group element for a specific color group within an SVG canvas, highlighting its role in organizing and displaying visual information.
- Explain how combining all color groups' SVG representations into one main canvas can streamline data presentation and improve overall visual clarity in an image processing project.

Output Generation for SVG

To finish the SVG picture, close the main SVG canvas.

Raster image conversion into SVG format has a variety of advantages for digital artists. SVG's intrinsic scalability and edit ability mean that pictures may be easily edited with vector graphics tools and enlarged without losing quality. Smaller SVG file sizes are better for sharing and storing, especially for graphics containing sections of consistent color. Automatic color segmentation assists in simplifying complicated artworks or assessing compositions. It's significant that the original raster picture is preserved, enabling artists to use both forms as necessary. The speed and creative freedom of the design process are

eventually improved by this seamless integration into vector operations, Zhang et al. (2019).

METHODOLOGY DIFFERENCE

The given approach combines conventional image processing methods with masking and SVG generation in a step-by-step procedure. This unique combination allows for precise manipulation of images while maintaining high quality output. The use of SVG generation also ensures scalability and compatibility across different devices and platforms.

Efficiency Difference

The efficiency of both strategies would rely on how they were specifically implemented. However, neural networks may require a lot of computer power and training data to recognize complicated patterns. This solution is based on well-known image processing methods, which may be more computationally effective in some situations, Ma et al. (2022).

Applicability Difference

The solution seems to be simple to use with current tools and libraries, making it accessible to a variety of people. However, it may be necessary to have knowledge of deep learning and specific neural network models for the "Neural SVG Decoder" (hypothetical). This level of expertise may limit the accessibility of the solution to a more specialized group of individuals. In order to bridge this gap, it would be beneficial to provide resources and training opportunities for those interested in utilizing the "Neural SVG Decoder." By offering workshops, online tutorials, and mentorship programs, we can empower a wider range of individuals to leverage this powerful tool in their own projects and research endeavors. Additionally, creating a user-friendly interface and documentation for the decoder can also help to lower the barrier to entry for those with less experience in the field.

RESULT & DISCUSSION

Taking image format whether in PNG or JPG the result in Figure 2 show that each group of layers are different from each other the simplest method of segmentation and then each group layer subtracted by original group layer which could be the simplest approach of converting raster file in SVG layers format.



Figure 2.1.
Output image on Illustrator

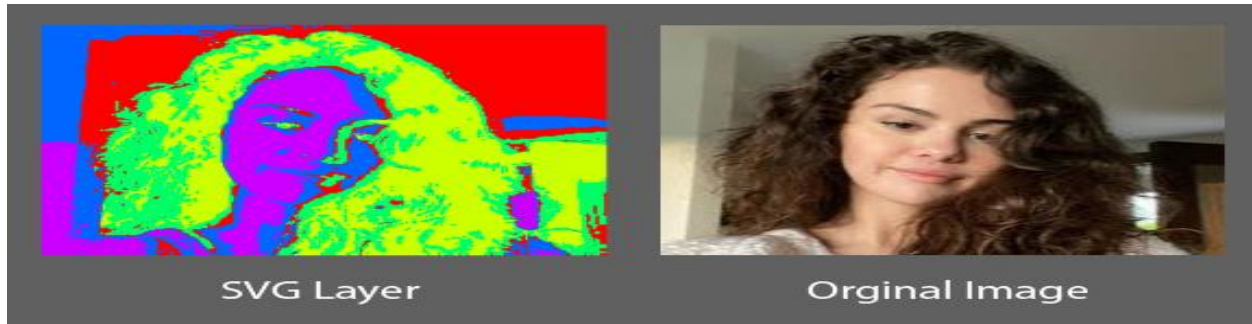


Figure 2.2.
Output image on Illustrator

It shows that each color group in Figure 4, is divided into a distinct layer for easy modification by anyone wishing to do so. With this method, the image's quality and resolution are maintained while being easily modified. It is noted that, as seen in Figure 3, a dataset of ten randomly selected actor images in format whether in png or in jpg was obtained to assess the algorithm's effectiveness. The results revealed that the pictures included a variety of performance metrics. Certain images showed remarkable recall, catching every positive pixel, while others showed good accuracy, avoiding false positives. Overall, the dataset provided a comprehensive evaluation of the algorithm's performance across different metrics. The diversity in the images allowed for a thorough analysis of its capabilities in detecting actors in various poses and lighting conditions. The results highlighted the algorithm's strengths in both sensitivity and specificity, indicating its potential for accurate and reliable actor detection in real-world scenarios.

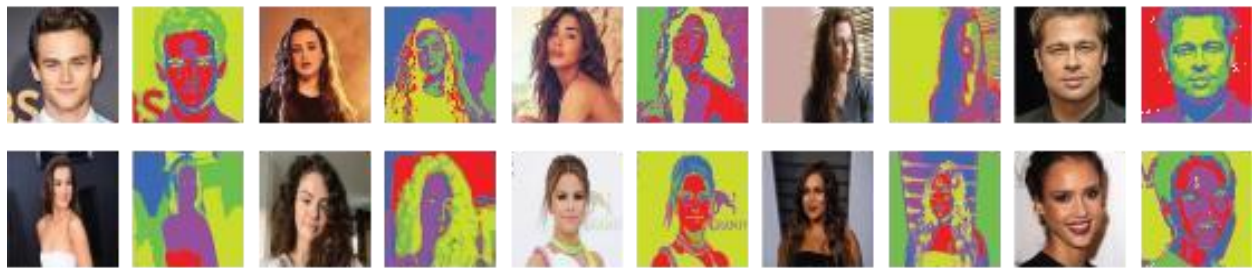


Figure 3.
Random 10 Images Dataset



Figure 4.1.
Color Group Layer

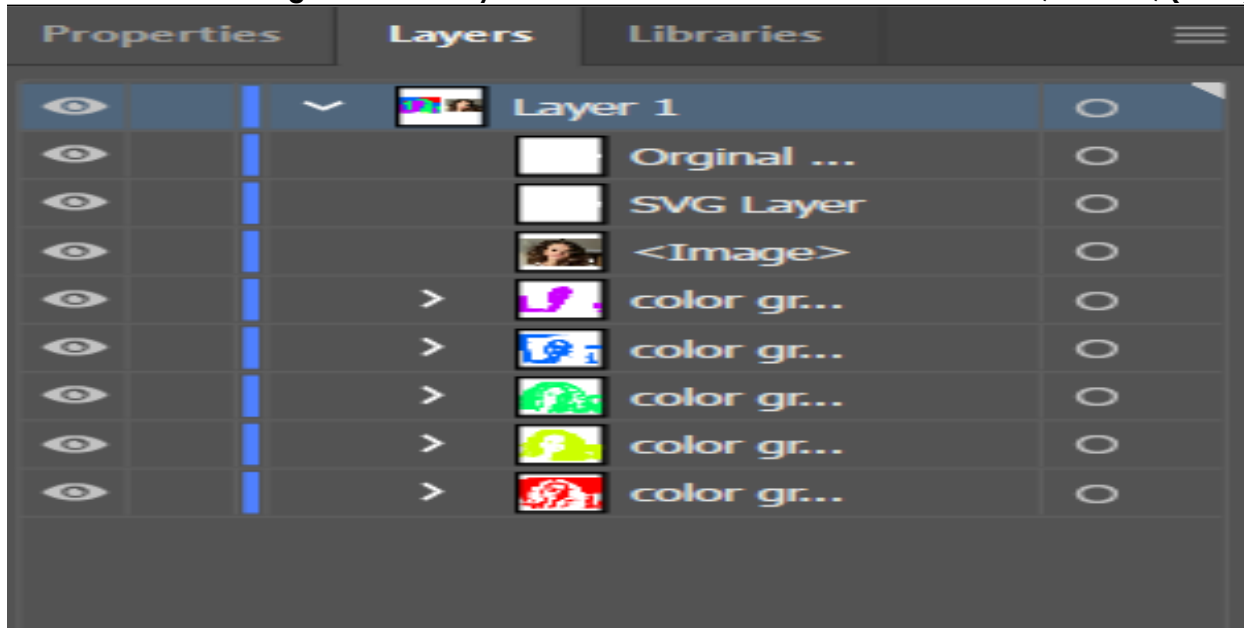


Figure 4.2.
Color Group Layer

Comparison

As compared to a sophisticated approach such as the concept of "Neural SVG Decoder" [1], employing a simple technique to transform images into SVG format highlights usefulness and practicality. The simplicity of the suggested clustering technique, which uses basic color clustering instead of complex image processing or neural networks, is noteworthy. This simple method reduces the work to just choosing an image and pressing the "Convert" button to produce an SVG output. Even though the simple method might not have all the advanced features, it is a quick and useful fix for users who value simplicity and speed over complex image processing and neural network-driven conversion.

More specifically, as a deep learning model, the Neural SVG Decoder offers sophisticated capabilities, but it also comes with complexity and challenges. It requires a large amount of training data and complex parameter tuning, and it can occasionally result in incomplete or erroneous SVG descriptions, especially when the generator network is not properly trained or is using poor-quality training data. In Figure 5, on the other hand, the suggested clustering approach excels in its simple to use and simplicity even though it lacks some of the more sophisticated features of the Neural SVG Decoder. Without having to delve into the complexities of deep learning, users can quickly convert images into SVG format, making it appropriate for a wider range of users, including designers and artists who value simplicity and efficiency.

Additionally, the suggested clustering technique focuses on transforming pictures into SVG layers so that edits can be made quickly and easily. This method increases the SVG format's adaptability by allowing digital artists to quickly alter or modify particular layers without sacrificing the image's overall quality.



Figure 5.
Result of same Image with Adobe Express (Neural SVG Decoder)

EXPLANATION

The first line of the code loads an input image from a file. Defining Colour Clusters: Users can choose how many color clusters (in this example, five) they wish to appear in the final image. Color Clustering: The code groups colors in the image that are similar by using a clustering method (KMeans). Then, it gives each pixel a place in a certain color cluster. It produces a color map for each of the different color groupings. It makes use of the HSV color space in this code. Creating an empty SVG canvas, which functions as a blank canvas on which to draw the final SVG picture, is the first step in creating an SVG file. Based on the provided picture, it determines the canvas's dimensions. Color Group Looping: The code iterates across each color group:

To separate the pixels that belong to that color group, it builds a mask. It applies the mask on the source image to produce a color-isolated version of it. This single image has been transformed into an SVG version. This color group receives its own group element in the SVG. The SVG for the color group is added to the main SVG canvas. The main SVG canvas is closed once all color groups have been processed, which completes the SVG. SVG Image Saving: The completed SVG image is saved to a designated output file. Displaying and Saving the Result: A message verifying the generation of the SVG image and its filename is displayed by the code.

The saved SVG image can now be easily displayed on a web page or used in various design projects. This process simplifies the conversion of images into scalable vector graphics for better quality and flexibility. Hexadecimal conversion: Auxiliary functions are supplied to translate RGB color data into hexadecimal representation. In short, this code takes an input image, group's colors that are similar, and creates an SVG graphic with groups based on colors. It's a helpful tool for condensing photos into the scalable and editable SVG format, which makes it perfect for usage in a variety of digital art and design applications. Additionally, SVG files can be easily manipulated and customized

without losing quality, making them ideal for web design and animation projects. They also have smaller file sizes compared to other image formats, resulting in faster loading times on websites.

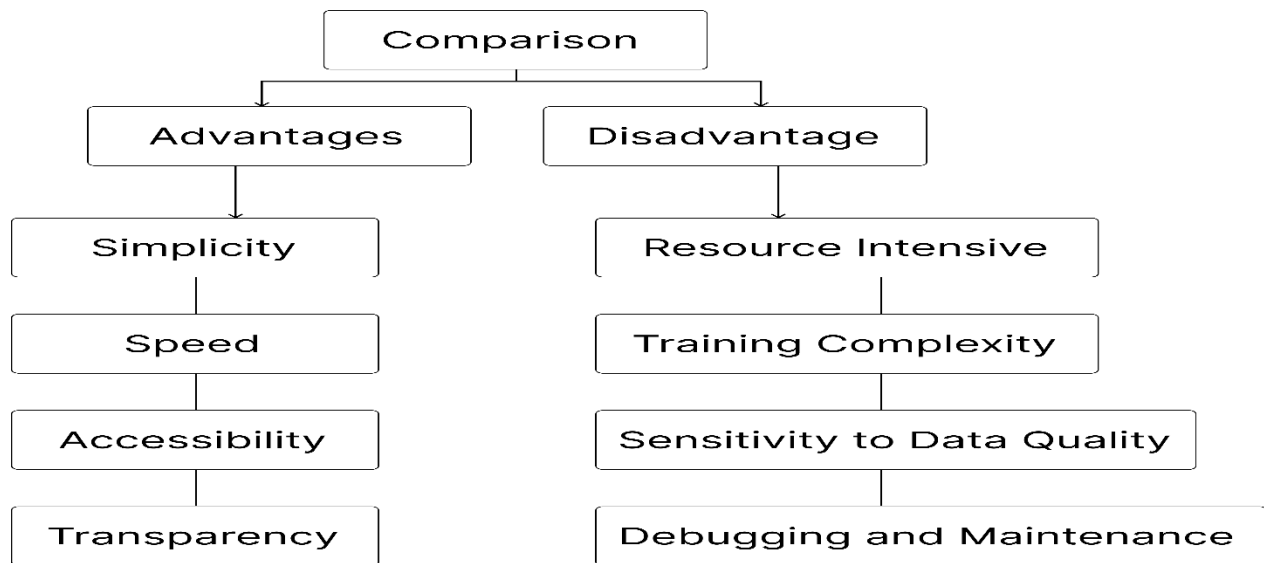


Figure 6. Cluster of Advantages of Proposed Method & Disadvantages of Neural SVG Decoder

Advantages

Simplicity: By utilizing color clustering techniques, the suggested clustering method provides a more straightforward method for converting raster’s to vectors.

Speed: This method can be faster than intricate neural network-based techniques by segmenting the image into color clusters and producing vector images for each cluster.

Accessibility: A wider audience may find the clustering method more suitable as it may be more understandable for users lacking in-depth knowledge of deep learning.

Transparency: Compared to deep learning models, the processes of color clustering and vector image generation are more apparent and simpler to comprehend.

Disadvantages

Resource-intensive: A substantial amount of processing power and training data are needed to train the neural SVG decoder. **Training Complexity:** Deep learning techniques are required, and the training process can be both time-consuming and complex. **Sensitivity to Training Data:** The caliber and variety of the training data have a significant impact on the model's performance. Results could be less than ideal if the data is biased or insufficient. **Debugging and Maintenance:** The model's complexity makes it difficult to debug and improve, necessitating constant parameter monitoring and adjustment

Test on Different Images

Given Algorithm applied 10 images (as shown in Table) of different gender with different color to find out result accuracy. For different images, the given dataset includes

precision, recall, and F1-Score values in Table 1. These metrics are frequently used to evaluate how well binary classification or image algorithm are working. The precision value indicates the percentage of correctly predicted positive observations out of all positive predictions made. Recall, on the other hand, measures the percentage of actual positives that were correctly identified by the algorithm.



Figure 7.
Block Diagram of Algorithm

Precision and recall are taken into consideration while calculating the F1-Score Abro et al. (2020), which is a balanced performance metric. The findings indicate different levels of success in svg conversion algorithm for various images, with some achieving a reasonable balance between precision and recall and others maybe needing improvement in one or both of these parameters. Depending on the particular application and the trade-offs between false positives and false negatives, the most crucial measure is determined. This analysis highlights the importance of fine-tuning svg conversion algorithms to suit the specific needs of different types of images. It also underscores the need for ongoing evaluation and refinement of these algorithms to ensure optimal performance in a variety of scenarios. Additionally, further research could explore the potential for machine learning techniques to enhance the accuracy and efficiency of svg conversion algorithms.

Table 1.

Data Set	Precision	Recall	F1-Score
----------	-----------	--------	----------

Convert Rasterize Image Into SVG Layers			Hassan, A et al., (2024)
Img1	0.70266	0.58545	0.63872
Img2	0.88981	1	0.94169
Img3	0.70121	0.84092	0.76568
Img4	0.39194	0.28263	0.32843
Img5	1	0.94645	0.97249
Img6	0.46193	0.31471	0.37437
Img7	0.67851	0.792182	0.73078
Img8	0.89128	0.928121	0.90926
Img9	0.95214	0.72909	0.82582
Img10	0.61872	0.92461	0.74135

Precision Matrix

CONCLUSION

In this research, we created an effective approach for converting images to the SVG format and tested its effectiveness on a variety of images using accuracy, recall, and F1-Score measures. The photos' findings show a variety of outcomes, showing both successful and difficult situations. We noticed a range of performance indicators in the images. While "Img5" shown good accuracy, avoiding false positives, images like "Img2" demonstrated exceptional recall, catching all positive pixels. On the other side, "Img4" has issues with recall and accuracy, highlighting areas for development. We have so many opportunities in this project improvement like we can use advance level of color clustering and thresholding, color mapping also we can work on the outline and curves of the image we get in SVG which I found one reference as well called SVG enhancement. We aim to give digital artists and researcher's strong tools for image alteration, editing, and imaginative expression as we continue to improve and develop. Our team is dedicated to pushing the boundaries of image processing and enhancing the capabilities of our software. By incorporating these advanced techniques and technologies, we are confident that we can elevate the quality and precision of our results. Additionally, by focusing on the finer details such as outline and curves, we can create images that are not only visually stunning but also technically sound. The potential for growth and innovation in this field is vast, and we are excited to explore all the possibilities that lie ahead.

DECLARATIONS

Acknowledgement: We appreciate the generous support from all the supervisors and their different affiliations.

Funding: No funding body in the public, private, or nonprofit sectors provided a particular grant for this research.

Availability of data and material: In the approach, the data sources for the variables are stated.

Authors' contributions: Each author participated equally to the creation of this work.

Conflicts of Interests: The authors declare no conflict of interest.

Consent to Participate: Yes

Consent for publication and Ethical approval: Because this study does not include human or animal data, ethical approval is not required for publication. All authors have given their consent.

REFERENCES

- Abro, A. A. (2020). Identifying the machine learning techniques for classification of target datasets. *Computer Science-Artificial Intelligence*.
- Abro, A. A., Khan, A. A., Talpur, M. S. H., Kayijuka, I., & Yaşar, E. (2021). Machine learning classifiers: a brief primer. *University of Sindh Journal of Information and Communication Technology*, 5(2), 63-68.
- Abro, A. A., Talpur, M. S. H., Jumani, A. K., Siddique, W. A., & Yaşar, E. (2021). Voting combinations-based ensemble: A hybrid approach. *Celal Bayar University Journal of Science*, 18(3), 257-263.
- Abro, A. A., TAŞCI, E., & Aybars, U. G. U. R. (2020). A Stacking-based Ensemble Learning Method for Outlier Detection. *Balkan Journal of Electrical and Computer Engineering*, 8(2), 181-185.
- Carlier, A., Danelljan, M., Alahi, A., & Timofte, R. (2020). Deepsvg: A hierarchical generative network for vector graphics animation. *Advances in Neural Information Processing Systems*, 33, 16351-16361.
- Jain, A., Xie, A., & Abbeel, P. (2023). Vectorfusion: Text-to-svg by abstracting pixel-based diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 1911-1920).
- Jiazheng, Y., Jinghua, H., Yujian, W., & Hong, B. (2008). Converting real images to SVG based on XML.
- Ma, X., Zhou, Y., Xu, X., Sun, B., Filev, V., Orlov, N., ... & Shi, H. (2022). Towards layer-wise image vectorization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 16314-16323).
- Nadeem, G., Rehman, Y., Khaliq, A., Khalid, H., & Anis, M. I. (2023, March). Artificial Intelligence based prediction system for General Medicine. In *2023 4th International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)* (pp. 1-6). IEEE.
- Schwab, M., Saffo, D., Bond, N., Sinha, S., Dunne, C., Huang, J., ... & Borkin, M. A. (2021). Scalable scalable vector graphics: Automatic translation of interactive svgs to a multithread vdom for fast rendering. *IEEE Transactions on Visualization and Computer Graphics*, 28(9), 3219-3234.
- Vun, N., Tee, Z. C., & Lee, J. Y. (2009, December). Enhancement of SVG display for embedded applications. In *2009 IEEE 9th Malaysia International Conference on Communications (MICC)* (pp. 68-72). IEEE.
- Wei, Q., Li, D., Liu, C., Liu, B., & Liu, Z. (2011). Converting digital image to svg for user interaction. In *Applied Informatics and Communication: International Conference, ICAIC 2011, Xi'an, China, August 20-21, 2011, Proceedings, Part I* (pp. 378-385). Springer Berlin Heidelberg.
- Zhang, X., Xu, Y., Lin, Q., Qiao, B., Zhang, H., Dang, Y & Zhang, D. (2019, August). Robust log-based anomaly detection on unstable log data. In *Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering* (pp. 807-817).



2024 by the authors; Asian Academy of Business and social science research Ltd Pakistan. This is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC-BY) license (<http://creativecommons.org/licenses/by/4.0/>).