SegPalette: Instance Recolouring by Palette Generation after Instance Segmentation

Brendan Bickford*
Simon Fraser University

Andrei Harbachov*
Simon Fraser University

Carolina Partida Bujanda*
Simon Fraser University

Sebastian Dille
Simon Fraser University

Yağız Aksoy
Simon Fraser University

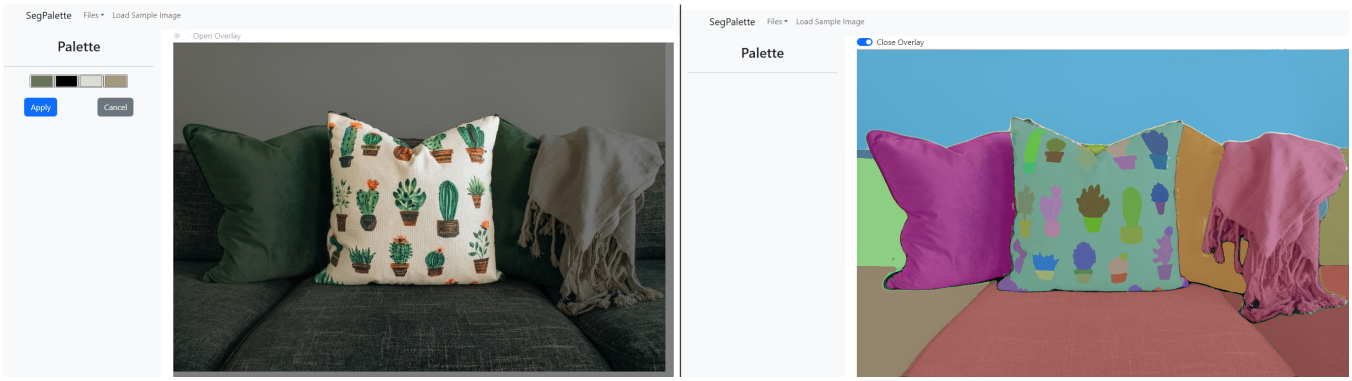

Figure 1: SegPalette Web App showing palette generation and image segmentation.

ACM Reference Format:
Brendan Bickford, Andrei Harbachov, Carolina Partida Bujanda, Sebastian Dille, and Yağız Aksoy. 2024. SegPalette: Instance Recolouring by Palette Generation after Instance Segmentation. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Image recolouring can be a daunting task for users who are not familiar with photo editing tools. Building a simple application to allow users to easily recolour their images has been the goal of multiple papers over the last decade leading to some great tools. The problem is that the existing tools such as the Palette-based Photo Recoloring from Princeton University [Chang et al. 2015] require the user to recolour the entire image all at the same time. This would make it impossible to recolour just a single object in an image without altering the rest of the image.

Our goal with SegPalette is the provide the user with a similar palette-based recolouring tool while allowing the selection of specific segments to be recoloured. To do this, SegPalette integrates the Segment Anything Model (SAM) from Meta AI [Kirillov et al. 2023] to segment an uploaded image. Additionally, the user can upload an existing segmentation file if they already have one.

Allowing the user to select specific segments of the image to recolour enables them to perform specific recolouring of objects. For instance, the user could change to colour of an outfit that someone is wearing in the image or change the colours of a decorative pillow. This segment-specific palette generation and recolouring process makes it possible for SegPalette to be used in fashion design, photo tourism and home decor applications.

2 METHOD

2.1 User Interface

Our user interface (UI) is designed to be simple and intuitive. Users can drag and drop or upload an image from their device. If the user does not have an image prepared, they can select the sample image from the navigation bar. Once the image is selected, the user can upload a segmentation file for the image, or use the built-in segmentation model. Once the image and segmentations are loaded, the user can simply click on the segment that they wish to recolour and the colour palette will be generated. Next, the user can click on the colour they wish to change and use the colour picker to modify it. Once all the colour changes are complete, the final image and segmentation file can be saved to the user's device.

*Denotes equal contribution.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

2.2 Image Segmentation

To isolate specific objects within the image for targeted recolouring, we use segmentation masks. The generation of segmentation masks is facilitated by two primary methods: using a pre-existing COCO segmentation file generated through any means or utilising the Segment Anything Vision Transformer Huge model. This model excels in producing highly precise segmentation masks suitable for diverse image scenarios.

After getting the segmentation data, the Run-Length Encoding (RLE) is decoded to produce separate binary masks. These binary masks are used for the delineation of segmented objects within the image to facilitate object-specific recolouring.

To optimise efficiency and streamline subsequent operations, the binary masks are multiplied against the original image to have them retain their corresponding pixel values from the original image, while pixels outside the segmentation objects are assigned a value of 0. By integrating the original image with the binary masks, there is no need to retrieve pixel values from the original image that is covered by the binary mask for each colour palette generation and recolouring operation, thereby enhancing overall efficiency and performance.

2.3 Colour Palette Generation

In order to customise the colour palette generation to the specified segmentation, we begin by taking a mask of the original image. The mask constrains the original pixel colours for pixels within the segmentation and the rest of the pixels are set to 0. When generating the colour palette, we use a k-means clustering algorithm with k=6. When initialising the cluster centroids, we ensure that only pixel colours from inside of the segmentation are selected from. This ensures that the palette is representative of the segmentation compared to a naive k-means implementation over the entire mask.

2.4 Recolouring

The recolouring process aims to modify the colour appearance of specific segments within an image while preserving their original luminance characteristics. This is achieved through the manipulation of colour channels within the LAB colour space.

To perform recolouring, the 'a' and 'b' channels of the LAB image are adjusted based on the provided colour palette for the segment. This adjustment involves calculating the difference between the original colour values and the target colour values from the palette. By applying this difference to the corresponding 'a' and 'b' channels, the colour appearance of the segment is modified accordingly, while accounting for the perceptual uniformity of the LAB colour space. By using LAB colour space for recolouring, our method provides users with a flexible and efficient tool for achieving desired colour transformations in their images.

The implementation of the recolouring feature wasn't completed for the web app. Therefore, we conducted our recolouring experiments using a MATLAB script to validate the feasibility of the theoretical framework.

3 RESULTS

In light of the recolouring feature not being implemented on the web app, we used MATLAB to test the recolouring and the theoretical feasibility of our framework overall. Reviewing the outcomes of the recolouring conducted in MATLAB (Figure 2), several key points emerge. Firstly, while recolouring a person's outfit (Figure 2a), most of the clothing transitioned smoothly, but a minor issue occurred where a portion of the lower lip received a green hue. Secondly, altering designs on a throw pillow (Figure 2b) led to some areas retaining the original colour due to segmentation network limitations, particularly noticeable given the small size of the designs. In contrast, recolouring a rug (Figure 2c) showed flawless results, likely due to its uniform colour and ample size, aiding precise segmentation. Lastly, recolouring the couch (Figure 2d) benefited from clearer and larger designs, allowing the segmentation model to accurately delineate the area, resulting in successful recolouring.

4 CONCLUSION

SegPalette offers a novel approach to instance recolouring by integrating segment-specific palette generation after instance segmentation. While our web app currently lacks recolouring functionality, successful results obtained via MATLAB affirm the feasibility of our theoretical framework. Our method enables users to selectively recolour individual segments within an image, providing versatility for various applications. Despite the successful recolouring showcased, we observed some limitations, such as minor colour bleeding and segmentation network constraints. Moving forward, implementing the recolouring on the web app and addressing these limitations could pave the way for creating a user-friendly recolouring tool with high usability. Overall, SegPalette presents a promising avenue for intuitive and precise image recolouring, opening possibilities for future research and development in this domain.

REFERENCES

Bala Baskar. 2022. Machu Picchu. Dataset. https://www.kaggle.com/datasets/balabaskar/wonders-of-the-world-image-classification/data

Huiwen Chang, Ohad Fried, Yiming Liu, Stephen DiVerdi, and Adam Finkelstein. 2015. Palette-based photo recoloring. *ACM transactions on graphics* 34, 4 (2015), 1–11.

Ksenia Chernaya. 2020. Brown Wooden Desk With Chair. Photograph. https://www.pexels.com/photo/brown-wooden-desk-with-chair-3952038/

Designecologist. 2018. White and Green Throw Pillows. Photograph. https://www.pexels.com/photo/white-and-green-throw-pillows-1248583/

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. 2023. Segment Anything. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)* (2023), 3992–4003.

Gustavo Peres. 2019. Two White and Green Floral Padded Sofa Chairs Next to Wall in Room. Photograph. https://www.pexels.com/photo/two-white-and-green-floral-padded-sofa-chairs-next-to-wall-in-room-1885562/

(a) Original

(a) Recoloured*

(b) Original

(b) Recoloured*

(c) Original

(c) Recoloured*

(d) Original

(d) Recoloured*

Figure 2: Comparison of original and recoloured images*. Original images (a), (b), (c), and (d) are from [Baskar 2022], [Designecologist 2018], [Chernaya 2020], and [Peres 2019], respectively. *Results were obtained from MATLAB scripts rather than the web app.