Sofa Image Extraction and Background Replacement for E-commerce

***Abstract*— In the competitive landscape of e-commerce, high-quality product images are essential for enhancing customer engagement and increasing sales. However, capturing furniture images in clutter-free environments remains a challenge for many sellers. This paper presents an image-processing-based pipeline for automated sofa image extraction and background replacement, eliminating the need for expensive studio setups or deep-learning models. Our method integrates color-based segmentation, K-Means clustering, morphological operations, and GrabCut refinement to effectively isolate sofas from complex backgrounds. The approach was tested on a dataset of 30+ images obtained from Wayfair, consisting of both simple and complex backgrounds. While achieving a 100% accuracy rate for simple backgrounds, the method demonstrated a 70% success rate in complex cases. The primary failure modes were observed in scenarios with low contrast and color similarity between the sofa and its surroundings. These results highlight the effectiveness of classical segmentation techniques while also underscoring their limitations. Future work could incorporate deep learning-based enhancements, such as Mask R-CNN or U-Net, to improve segmentation robustness and adaptability in challenging real-world conditions.**

Keywords*— *Image Segmentation, Background Replacement, Furniture Extraction

I. Introduction

In the era of e-commerce, high-quality product images play a crucial role in attracting customers and influencing purchase decisions. When selling furniture online—whether on platforms like Wayfair, IKEA, or vintage marketplaces—clean and professional product images significantly enhance the perceived value of the item. However, individuals and small businesses often struggle with capturing furniture images in clutter-free environments. Unwanted backgrounds, poor lighting, and inconsistent settings can make products look unappealing and unprofessional. This challenge highlights the need for an automated solution to extract furniture images from their original backgrounds and place them onto a clean, uniform backdrop, ensuring a polished and market-ready appearance.

This paper presents a purely image-processing-based approach to sofa image extraction and background replacement, eliminating the need for expensive studio setups or deep learning-based segmentation models. By leveraging techniques such as color-based segmentation, K-Means clustering, morphological operations, and GrabCut refinement, our method efficiently isolates furniture from complex backgrounds. This approach is beneficial not only for individual sellers looking to resell furniture but also for businesses that require large-scale product image enhancement. The proposed solution can be integrated into online marketplaces, furniture retailers, and vintage stores, ensuring consistent and high-quality visuals that improve customer engagement and sales potential.

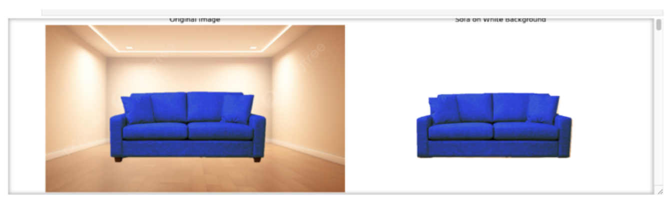

Fig 1. Sofa Extraction

II. Background

Image segmentation is a crucial task in computer vision, particularly for applications in e-commerce where accurate object extraction enhances visual appeal and user experience. Various traditional and deep learning-based methods have been explored to achieve precise segmentation.

Classical image segmentation techniques, such as thresholding, edge detection, and region-based methods, have been widely used for object extraction [1]. Watershed and superpixel-based segmentation approaches have shown effectiveness in delineating object boundaries, especially in complex backgrounds [2]. However, these methods often struggle with variations in lighting, shadows, and occlusions, making them less reliable for real-world applications.

Advanced machine learning and deep learning techniques, including U-Net and Mask R-CNN, have significantly improved segmentation accuracy by leveraging convolutional neural networks (CNNs) to learn hierarchical features [3]. U-Net has demonstrated exceptional performance in biomedical image segmentation and has been adapted for various object segmentation tasks due to its encoder-decoder architecture [4]. Mask R-CNN extends Faster R-CNN by incorporating a segmentation branch, enabling pixel-wise object mask predictions, which is particularly useful for detecting and extracting objects in cluttered environments [5].

Recent studies have explored hybrid approaches that integrate deep learning with classical methods to enhance segmentation robustness. For instance, the combination of object detection models such as YOLO or Faster R-CNN with traditional segmentation methods like GrabCut has shown promising results in refining extracted object boundaries [6]. This hybrid approach enables the detection of target objects followed by precise segmentation refinement, addressing challenges related to object occlusion and background clutter.

Although significant progress has been made in object segmentation, accurately extracting sofas from complex backgrounds remains a challenge. Conventional techniques such as thresholding and region-based segmentation often fail to produce precise object boundaries, while deep learning models like Mask R-CNN and U-Net require substantial labeled datasets and high computational resources. To address these limitations, our method combines color-based segmentation, K-Means clustering, and GrabCut refinement, improving segmentation accuracy while remaining computationally efficient. By utilizing center pixel color detection and adaptive segmentation strategies, our approach enhances sofa extraction without the need for extensive training data. This hybrid technique bridges the gap between traditional and deep learning-based methods, making it a practical solution for e-commerce applications that demand both accuracy and efficiency.

III. Methodology

A. Overview

The objective of this project is to extract sofas from complex backgrounds using a combination of classical image processing and machine learning-based segmentation techniques as shown in Fig 1. Given the challenges of background clutter and varying lighting conditions, a hybrid approach was employed that integrates color-based segmentation, clustering, and iterative mask refinement.

The pipeline follows a multi-step process:

1. **Image Preprocessing** – The image is loaded, converted to RGB and grayscale formats for further processing.
2. **Center Pixel Color Detection** – Identifies the dominant sofa color to assist in segmentation.
3. **K-Means Clustering** – Groups similar color regions to segment the sofa.
4. **Sofa Mask Extraction** – Uses clustering results to define an initial segmentation.
5. **GrabCut Refinement** – Improves segmentation using an iterative foreground-background separation algorithm.
6. **Mask Refinement** – Applies morphological operations and contour filtering to enhance the segmentation.
7. **Final Extraction** – Places the sofa on a clean white background.

This combination of techniques allows for efficient and accurate object extraction without requiring large labeled datasets, making it suitable for e-commerce applications.

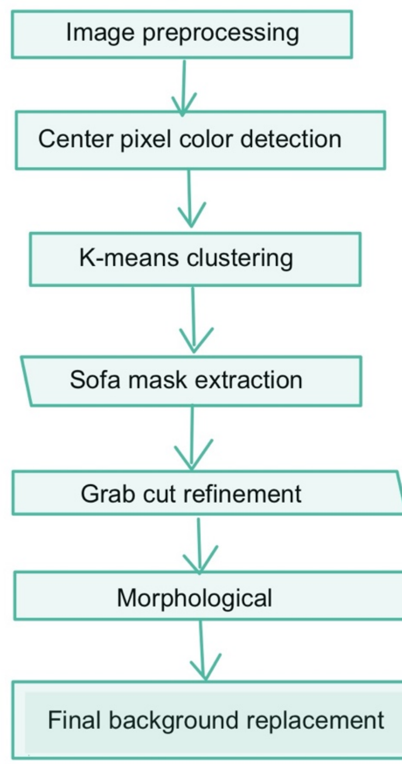

Fig 2. Sofa Extraction Pipeline

B. Pipeline

1. Image Preprocessing

The first function in the pipeline, load_image(image_path), is responsible for reading and loading the input image. Using OpenCV's cv2.imread(), the image is loaded into memory in its original format. It is then converted into two formats: RGB and Grayscale using cv2.cvtColor(). The RGB format is necessary for color-based processing later in the pipeline, while the Grayscale version is helpful for potential future enhancements like edge detection or thresholding.

2. Center Pixel Color Detection

The get_center_pixel_color(img_rgb) function focuses on determining the dominant color of the sofa by analyzing the pixel in the center of the image. This function works by extracting the RGB value of the center pixel and converting it to the HSV color space using cv2.cvtColor(). The resulting HSV values are used to define an appropriate range for color segmentation, helping to isolate the sofa's color in the next stages of the pipeline. The function's key role is identifying a color range that will be most useful for detecting the sofa amidst other background elements.

3. K-Means Clustering for Segmentation

In apply_kmeans_segmentation(img_rgb, num_clusters=7), the image undergoes a segmentation process using K-Means clustering. First, the image is reshaped into a list of pixel values, which are then clustered into a predefined number of color clusters (7 by default) using cv2.kmeans(). This results in a segmented version of the image, where each pixel is assigned to a particular cluster based on its color. This step helps break down the image into regions of similar colors, facilitating the next phase of isolating the sofa.

4. Sofa Mask Extraction from Clustering Results

After segmentation, extract_sofa_mask(clustered_img) identifies the most dominant color cluster in the center of the image, where the sofa is expected to be located. By using np.unique() and np.argsort(), the function identifies the most common cluster in the center area and creates a binary mask. In this mask, pixels belonging to the dominant cluster are marked as foreground (255), while the rest are marked as background (0). This mask helps isolate the region of interest, which is the sofa, from the rest of the image.

5. GrabCut Refinement

The apply_grabcut_on_color_mask(img_rgb, sofa_mask, lower_bound, upper_bound, iter_count=20) function refines the initial mask obtained from K-Means clustering using the GrabCut algorithm. GrabCut is an iterative algorithm that requires a bounding box to initialize the segmentation. The bounding box is defined around the identified sofa region. Once the initial mask is provided, the algorithm iteratively separates the foreground (sofa) from the background. Afterward, morphological operations are applied using cv2.morphologyEx() to clean up any noise in the mask, enhancing the accuracy of the extraction.

6. Mask Refinement

The refine_mask(binary_mask) function applies morphological operations, specifically opening and closing, to the binary mask. This smooths the edges and removes small background artifacts that could interfere with the segmentation. Additionally, contours are identified using cv2.findContours(), and the largest connected component, which corresponds to the sofa, is isolated. This step is essential for ensuring that the final mask cleanly represents the sofa without unnecessary background elements.

7. Final Extraction on White Background

In extract_sofa_on_white_bg(img_rgb, final_mask), the refined mask is applied to extract the sofa from the original image. All non-sofa pixels are set to white ([255, 255, 255]), creating an image where only the sofa remains, with a clean white background. This step ensures that the extracted sofa is ready for presentation or further processing, with a uniform and distraction-free background.

8. Displaying Results

The display_images(images_list, titles_list) function allows for the visualization of the results at each step of the pipeline. Using Matplotlib's plt.imshow() and plt.subplot(), this function displays a side-by-side comparison of the images at different stages, making it easier to debug or analyze the results. Depending on the display_type parameter, users can choose to see either all intermediate steps for debugging (display_type=1) or just the original and final images for a more streamlined comparison (display_type=2).

9. Complete Pipeline Execution

The final function, sofa_extraction_pipeline(image_path, display_type=1), orchestrates the entire workflow. It sequentially calls all the previous functions, ensuring that each step is performed in the correct order. The display_type parameter allows for flexible output, showing either all steps for debugging or just the original and final results. This function ties together the preprocessing, segmentation, refinement, and visualization steps into a cohesive pipeline that can be applied to any given input image.

Overall, this image processing pipeline efficiently extracts sofas from images, making it highly valuable for e-commerce platforms requiring background removal and product isolation.

Fig 3. Sofa Extraction Pipeline Results

IV. Results

To evaluate the effectiveness of the proposed image segmentation pipeline for furniture image extraction, we conducted experiments using a dataset of sofa images obtained from the Wayfair website. The dataset consists of 30+ test cases with varying background complexities and contrast levels. Specifically, the dataset includes 7 images with simple backgrounds and 23 images with complex backgrounds, designed to test the robustness of our approach in real-world e-commerce scenarios. The simple backgrounds primarily feature plain walls or solid-colored surfaces, while the complex backgrounds contain various elements such as indoor decorations, books, leaves, other furniture pieces, and window panes. Additionally, some images exhibit low contrast between the sofa and its surroundings, further challenging the segmentation process.

The segmentation performance was evaluated based on the accuracy of furniture extraction and background replacement with a clean white backdrop, which is a crucial requirement for e-commerce product listings. For the 7 images with simple backgrounds, the model demonstrated a 100% accuracy rate, successfully extracting the furniture without any noticeable artifacts. The well-defined edges and contrast in these images contributed to the reliable performance of the segmentation techniques. The extracted sofas were cleanly separated from the background, making them suitable for direct use in product catalogs.

However, in the 23 test cases with complex backgrounds, the performance varied significantly. Out of these, 16 cases were successfully segmented, while 7 cases exhibited failure modes, resulting in an overall success rate of approximately 70%. The failure cases were analyzed to identify the primary causes of missegmentation. One of the major issues occurred when green sofas were placed in front of backgrounds containing green plants Fig 5. The similarity in color and texture between the foreground and background caused the segmentation algorithm to misclassify parts of the furniture as background, leading to incomplete extraction. Fig 4. Another failure mode was observed with leather sofas that had strong light reflections. The specular highlights on the glossy surface of the leather material introduced inconsistencies in the edge detection process, causing errors in object boundaries. Additionally, low-contrast scenarios, where the sofa had a similar color to the surrounding background (e.g., beige couches against off-white walls), resulted in poor segmentation performance due to the lack of distinct edges for separation.

These findings highlight the strengths and limitations of the current pipeline. While it performs well in controlled conditions with simple backgrounds, real-world complexities such as color similarity and lighting effects present significant challenges. Future improvements could involve integrating deep learning-based approaches like Mask R-CNN or U-Net, which could potentially learn contextual differences between the foreground and background, improving segmentation robustness. Furthermore, post-processing techniques such as edge refinement and contrast enhancement could help mitigate errors in low-contrast scenarios.

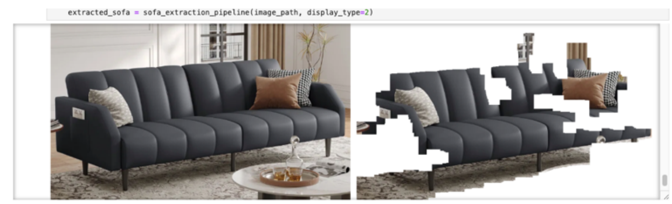

Fig 4. Sofa Extraction Pipeline Results

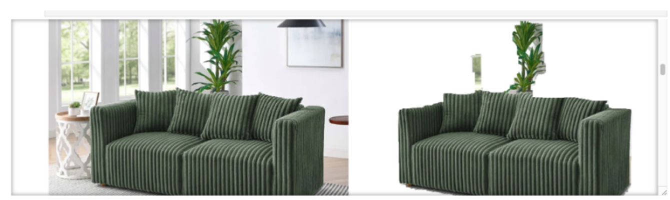

Fig 5. Sofa Extraction Pipeline Results

V. Conclusion

This project demonstrated the effectiveness of classical image processing techniques in extracting sofas from various backgrounds, particularly in controlled environments. The pipeline successfully segmented and replaced backgrounds for simple cases with 100% accuracy. However, performance dropped to 70% in complex scenarios, primarily due to color similarity between the sofa and background, reflections on leather surfaces, and low contrast conditions. These findings emphasize the strengths of our hybrid segmentation approach, particularly its computational efficiency and independence from large labeled datasets. However, the limitations suggest that incorporating deep learning-based segmentation models could improve performance, especially in complex and ambiguous cases. Future work could explore integrating Mask R-CNN for more robust object detection, leveraging contrast enhancement techniques, or employing adaptive segmentation strategies to refine extraction quality further. Despite its limitations, the proposed method provides a practical and accessible solution for improving e-commerce product imagery, offering a balance between accuracy and computational efficiency.

References

[1] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62-66, 1979.

[2] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167-181, 2004.

[3] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, 2015, pp. 234-241.

[4] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2961-2969.

[5] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[6] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Computer. Vis.*, 2015, pp. 1440-1448.