

Panoptic Segmentation for Indoor Environments using MaskDINO: An Experiment on the Impact of Contrast

Khalisha Putri* and Ika Candradewi⁺

* *Department of Computer Science and Electronics, FMIPA, Universitas Gadjah Mada, sekip utara BLS 21, Yogyakarta, Indonesia*

⁺ *Department of Computer Science and Electronics, FMIPA, Universitas Gadjah Mada, sekip utara BLS 21, Yogyakarta, Indonesia*

Received 18th of March, 2024; accepted 7th of August 2024

Abstract

Robot perception involves recognizing the surrounding environment, particularly in indoor spaces like kitchens, classrooms, and dining areas. This recognition is crucial for tasks such as object identification. Objects in indoor environments can be categorized into "things," with fixed and countable shapes (e.g., tables, chairs), and "stuff," which lack a fixed shape and cannot be counted (e.g., sky, walls). Object detection and instance segmentation methods excel in identifying "things," with instance segmentation providing more detailed representations than object detection. However, semantic segmentation can identify both "things" and "stuff" but lacks segmentation at the object level. Panoptic segmentation, a fusion of both methods, offers comprehensive object and stuff identification and object-level segmentation. In implementing indoor panoptic segmentation, considerations need to be made regarding the variabilities of room conditions, one of which is contrast. High or low contrast in the room potentially reduces the clarity of the shape of an object, thus affecting the segmentation results of that object. We experimented with how contrast varieties impact the panoptic segmentation performance using the MaskDINO model, the top model on the panoptic quality (PQ) leaderboard. We then improved the model generalization on the various contrasts by re-optimizing it using a contrast-augmented dataset, resulting in impressive outcomes with a PQ score of 47.7 %, a Recognition Quality score of 56.2%, and a Segmentation Quality (SQ) score of 76%.

Key Words: Panoptic Segmentation, MaskDINO, Indoor Environment, Contrast Enhancement.

1 Introduction

The rapid development of AI and robotics is increasingly evident over time. Robots in indoor environments are becoming more common, both in public spaces and within homes. Therefore, there is a need for robot perception, which involves robot navigation and AI technology in computer vision. Panoptic segmentation is used to produce detailed representations of each object within a space, which, when robust, can be utilized in various fields such as tourism as room service assistants, culinary fields as restaurant servers, household environments as cleanliness assistants, and many more. However, in its implementation, there are several aspects to consider, one of which is the condition of the room. The conditions of the rooms that a model will

Correspondence to: <ika.candradewi@ugm.ac.id>

Recommended for acceptance by Angel D. Sappa

ELCVIA ISSN:1577-5097

<https://doi.org/10.5565/rev/elcvia.1861>

Published by Computer Vision Center / Universitat Autònoma de Barcelona, Barcelona, Spain

face will not always be the same. Varied contrast conditions in rooms can affect the model's performance in defining object boundaries within the space. The term "contrast" refers to the differences in lighting and color among objects in an image. Images with higher contrast levels generally exhibit more significant color variation than those with lower contrast.

A dataset covering indoor objects is needed to implement panoptic segmentation in indoor environments. The COCO [1] dataset is one of the datasets that is commonly used to benchmark panoptic segmentation performances. MaskDINO [2] achieved first place on panoptic segmentation leaderboard [3] with a PQ of 59.5% using the Swin-L backbone, followed by k-Max Deeplab [4] of 58.5%, Mask2Former [5] with a PQ of 58.3%, Panoptic Segformer [6] with a PQ of 56.2%, and many other architectures. Based on the leaderboard rankings, transformer-based architectures dominate the top positions. Transformer-based architectures utilize attention mechanisms that selectively focus on the input sequence's essential parts. While these architectures outperform others, the attention mechanisms used to measure attention across all classes and objects require more computation. Therefore, based on the available leaderboard data, the details of implementing panoptic segmentation are as follows:

1. We filtered data from the COCO dataset to filter images based on specific categories and a predetermined size of 640x480 pixels.
2. We collected data for additional categories: plugs, toys, boxes, trash bins, fire extinguishers, plates, pans, stoves, and wallets. We annotated them according to the COCO Panoptic annotation format.
3. We implemented panoptic segmentation using the MaskDINO architecture by adjusting and tuning hyperparameters with the available computational resources.

After optimization, a study on the influence of contrast on the segmentation model's capabilities was conducted. The proposed steps for the experiment are as follows:

1. The model will be evaluated using validation data with contrast settings decreased by 50% and increased by 50% sequentially, and changes in its PQ, SQ, and RQ will be observed.
2. Then, we proposed to improve the model's generalization with two options for comparison: contrast enhancement and contrast augmentation, which will be applied in both the training and evaluation phases.
3. We also conducted model testing on images with varying contrasts and images of indoor objects, the categories specified in this study, to assess the model's ability to handle different room contrast conditions and its capability to segment and recognize objects within the space.

Overall, the model's ability to generalize object recognition under various contrast conditions is crucial in enhancing recognition quality in panoptic segmentation, helping the model detect, classify, and distinguish various objects more accurately and efficiently. Our research results showed an increase in Recognition Quality (RQ) and Panoptic Quality (PQ) values compared to the baseline model. Consequently, the implication is that optimizing the panoptic segmentation model can be implemented in robot perception to recognize objects under various contrast conditions. The model can be integrated into object-based Simultaneous Localization and Mapping (SLAM) within visual SLAM, helping robots interact with objects around them and assisting in autonomous navigation.

The rest of the research paper is structured as follows—section 2 covers related work, including panoptic segmentation and contrast enhancement methods. Section 3 introduces a proposed dataset for indoor objects. Section 4 outlines the proposed methodology for the research workflow. Section 5 details the experiments and results, along with a comparison of the suggested strategies for model improvement. Finally, Section 6 presents the research conclusions and discusses future directions.

2 Related Works

According to previous research about panoptic segmentation, two common types of architecture are used in panoptic segmentation: hybrid networks and transformer-based architectures. Hybrid architectures combine semantic segmentation and instance segmentation networks to segment stuff and things classes simultaneously. These architectures often incorporate feature fusion modules and multi-task learning approaches to optimize both segmentation tasks jointly. On the other hand, transformers were introduced by [7] as attention-based building blocks. The attention [7] mechanism is a neural network layer that merges information from all the input sequences to capture long-range dependencies and global context information, potentially improving performance on panoptic segmentation tasks.

Hybrid architectures are implemented on PanopticFPN [8], Panoptic-Deeplab [9], EfficientPS [10], REFINE [11], Panoptic FCN [12], and K-Net [13], which results in the PQ of 40,3%, 41,2%, 63,9% (on Cityscapes dataset), 51,5%, and 55,2% sequentially. DETR first utilized transformer in panoptic segmentation [14] model, which has inspired other models: MaskFormer [15], Panoptic Segformer[6], Mask2Former [5], OneFormer [16], kMaX-Deeplab [4], and MaskDINO [2], which result in the PQ of 45,1%, 53,3%, 56,2%, 58,3%, 58,5%, and 59,5% sequentially. The top rankings on the panoptic segmentation leaderboard are mostly transformer-based architectures. MaskDINO [2] performs best, but the model has not achieved SOTA for large-scale feature settings. kMax-Deeplab [4] can segment small objects in complex scenes but faces a challenge when segmenting heavily occluded objects and small objects that are not clear. OneFormer [16] can cut training time to three times due to its ability to segment images universally (semantic, instance, and panoptic segmentation). However, misprediction commonly happens across different segmentation tasks. MaskFormer [15] and Mask2Former [5] are accessible to users with limited computation resources. However, MaskFormer sometimes fails to detect commonly found objects (such as persons), and the Mask2Former model still finds it difficult to segment small objects. Panoptic Segformer[6] can still not face more extensive special features and small objects in images.

On the other hand, hybrid architectures tend to be in the lower rankings for panoptic segmentation leaderboards [3]. K-Net [13] can surpass the Panoptic Segformer and MaskFormer. However, due to its limited kernel, the model still finds it difficult to segment content with similar textures and all object instances. Panoptic FCN [12] is computationally efficient, as it avoids the need for fully connected layers and can process images of arbitrary sizes in a single forward pass. Nevertheless, the architecture typically involves downsampling operations that reduce spatial resolution, potentially losing fine-grained details in the segmented output. REFINE [11] produces better consistency between instance and semantic segmentation, fewer occlusion error estimations, and fewer false positive predictions. However, the model computation is relatively heavy, and its runtime is relatively slow. EfficientPS [10] is the most efficient model that achieves inference speed almost in real-time.

Nevertheless, like other non-transformer-based architectures, it has limitations in capturing long-range dependencies or global context information, particularly in complex scenes. It sometimes also fails to segment images with high contrast. Panoptic-Deeplab [9] can also achieve inference speed in almost real-time. However, the model still needs post-processing to achieve the final panoptic segmentation result. Panoptic FPN [8] is the first baseline for panoptic segmentation that is considered efficient due to its FPN architectures.

We proposed the panoptic segmentation research based on MaskDINO due to its highest performance on the COCO [1] dataset while considering our computational memory constraints. The architecture weakness, which is the expensive computational cost, can be mitigated by adjusting the training hyperparameter for the model. MaskDINO [2] is an integrated Transformer-based framework for object detection and image segmentation. It represents a natural extension of DINO, transitioning from detection to segmentation with minimal modifications to some essential components. Contrast enhancement improves the visual distinction between different elements in an image by adjusting the brightness, color, or intensity difference.

Contrast enhancement can be achieved through various methods, such as histogram equalization, contrast-limited adaptive histogram equalization, contrast stretching, gamma correction, local contrast enhancement, and unsharp masking. Histogram equalization effectively enhances the overall contrast of the image. It is simple

and easy to implement because it requires no parameter adjustments. However, it may amplify noise in regions with low contrast. Contrast-limited adaptive histogram equalization preserves local contrast better than global histogram equalization, thus making it practical for enhancing details in images with varying illumination.

Several methods are available for image contrast enhancement, each with specific advantages and challenges. Histogram equalization is a simple technique that enhances overall contrast without requiring parameter adjustments. However, it may amplify noise in low-contrast regions. Contrast-limited adaptive histogram equalization (CLAHE) improves local contrast preservation, particularly for images with uneven illumination, but it can over-amplify noise and artifacts in areas with high contrast variations [17]. Moreover, computing histograms and performing equalization for each local region is computationally intensive.

Contrast stretching enhances low-contrast images by expanding their intensity range, but it is unsuitable for images that already have sufficient contrast. Gamma correction effectively addresses non-linear intensity variations, improving contrast for images displayed on non-linear devices. However, careful tuning of the gamma parameter is essential, as excessive application may introduce artifacts and distortions.

Local contrast enhancement dynamically adjusts contrast based on regional characteristics, preserving details in bright and dark areas. While this approach is highly effective, it requires more computational resources due to its complexity compared to global methods. Unsharp masking sharpens edges and enhances details, improving perceived sharpness and contrast. However, it may produce halos and artifacts around edges if the sharpening radius is too large and is less effective for images with subtle or low-contrast details.

Each method has its strengths and limitations, and their suitability depends on the specific requirements of the application and computational constraints.

2.1 Dataset Benchmark

This section describes the dataset that includes indoor environments used for the experiment. MS COCO (Common Objects in Context) is a large-scale dataset containing 328,000 images of everyday objects and humans for object detection, segmentation, and captioning. COCO consists of 133 classes, including 80 thing classes and 53 stuff classes grouped into several supercategories. Scene categories that include indoor environments are: library, child's bedroom, church, dining room, office, auditorium, restaurant, shop, kitchen, house, living room, hotel, bathroom, classroom, market, factory, cafeteria, campus, hospital room, bedroom, food court, and plaza [1].

3 Proposed Indoor Environment Dataset

This section discusses the proposed dataset development used for the research. The dataset used for this research is a custom dataset that combines the COCO Panoptic 2017 dataset with additional images collected from phone recordings and various internet sources. This mixed-source dataset was specifically designed to reflect diverse indoor environments, providing a robust foundation for training the panoptic segmentation model. The dataset includes a broad range of indoor objects and categories, such as furniture, electronics, and other household items, with a total of 7542 training images, 2756 validation images, and 400 test images. The inclusion of indoor-specific categories, including plugs, stoves, and fire extinguishers, enhances the dataset's relevance for real-world applications in indoor robotic perception.

We captured and recorded several indoor rooms containing indoor objects in the predetermined categories using a high-quality phone camera from different viewing angles with a fixed scale of 4:3 ratio: bedroom, study room, and classroom. We also collected images of indoor rooms and objects from many internet sources, the dataset was further refined through filtering to ensure a consistent image size of 640x480 pixels, and annotations were created in the COCO Panoptic format. This not only ensured compatibility with existing segmentation models but also allowed for a detailed, object-level segmentation across both "things" (countable objects) and "stuff" (unstructured background elements). The addition of images from diverse sources like smartphones

and the internet helped to capture variations in lighting, perspectives, and object occlusions, simulating the variability encountered in real indoor environments.

Moreover, the dataset's segmentation annotations were provided in both mask and bounding box formats, enabling a comprehensive approach to panoptic segmentation, where both object recognition and detailed pixel-level segmentation are performed simultaneously. This combination of diverse categories, annotated data, and multi-source images made the dataset highly capable of supporting the panoptic segmentation model's ability to generalize across different room conditions, lighting situations, and object types, which is critical for real-world deployment.

3.1 Image Annotation

The proposed dataset annotation follows the COCO Panoptic annotation format, where each per-image annotation should have two parts: (1) a PNG that stores the class-agnostic image segmentation and (2) a JSON struct that stores the semantic information for each image segment. Data taken from COCO Panoptic 2017 already has an appropriate annotation format. On data taken from the camera and the internet, images will be labeled using the innovative polygon feature in the Roboflow application according to predetermined categories. The results of this labeling will produce annotation information summarized in labelme annotation format.

The labelme annotation format generated per image will be processed in two stages: converted into COCO Panoptic annotation format and separated based on object categories, things, and stuff, producing two labelme annotations for each image. The results of the annotation conversion to COCO Panoptic format will be combined into the filtered annotations from the COCO Panoptic 2017 dataset. Then, each annotation is separated into things and stuff object annotations, which will be converted into a PNG mask annotation, where the conversion rules for the two types of objects will be different. After generating two PNG mask images for each image data, the two images are combined using a bitwise function to produce a PNG mask image for panoptic segmentation.

Table 1: *COCO Panoptic JSON Annotation Format Key-Value Pairs Details.*

Key	Value	Example										
images	<p>The list of information of each image contains several relevant information in key-value pairs</p> <table border="1"> <thead> <tr> <th>Key</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>file_name</td> <td>File name of the image in string</td> </tr> <tr> <td>height</td> <td>Height of the image in integer</td> </tr> <tr> <td>width</td> <td>Width of the image in integer</td> </tr> <tr> <td>id</td> <td>Converted file name into an integer</td> </tr> </tbody> </table>	Key	Value	file_name	File name of the image in string	height	Height of the image in integer	width	Width of the image in integer	id	Converted file name into an integer	<pre>"images": [{ "file_name": "000000522418.jpg", "height": 480, "width": 640, "id": 522418 }, { "file_name": "000000309022.jpg", "height": 480, "width": 640, "id": 309022 }]</pre>
Key	Value											
file_name	File name of the image in string											
height	Height of the image in integer											
width	Width of the image in integer											
id	Converted file name into an integer											

Tabel 1 COCO Panoptic JSON Annotation Format Key-Value Pairs Details (continued)			
Key	Value	Example	
anno- tation	A list of annotations information of each image contains several relevant information in key-value pairs	<pre>"annotations": [{ "Segments_info": [list of segmentation information specified in the following table] ,"File_name": "000123.jpg" "Image_id": 123, }, { "Segments_info": "000000309022.jpg", "File_name": "000125.jpg", "Image_id": 125, }]</pre>	
	Key		Value
	Segments_info		The list of segmentation information of each object in the image contains several key-value-pairs
	File_name		File name of the image in string
	Image_id	Converted file name into an integer	
cate- gories	The list of categories information of each category contains several pieces of information in key-value pairs	<pre>"categories": [{ "supercategory": "person", "isthing": 1, "id": 1, "name": "person" }, { "supercategory": "furniture", "isthing": 1, "id": 65, "name": "bed" }, { "supercategory": "wall", "isthing": 0, "id": 171, "name": "wall-brick" }]</pre>	
	Key		Value
	Super category		Supercategory of the category in string
	isthing		Whether the category is things or stuff in a boolean
	id		Category id in integer
	name	Category name in string	

The annotation format are shown in Table 1. The annotation information contains several nested key-value pairs specified in the Table 2 For training MaskDINO, instances annotation is also required, which can be gained from COCO Panoptic 2017 from COCO Instances 2017. For the images captured from a phone and collected from the internet, the annotation process utilizes labelme annotations for things only, which are converted into COCO Instances annotation format and merged with the filtered annotation from COCO Instances 2017. The annotation stages are as follows:

1. The images are annotated in polygon.
2. Each instance of 'things' object segments on an image is assigned distinct colors for PNG annotations.
3. Each 'stuff' object segment is assigned to the same color for the same category. Each stuff category is represented in a different color.
4. For a group of objects that belongs to the same category (e.g., a pile of books relevant to the things category only), set the iscrowd value in the JSON annotation to 1.

4 Proposed Methodology

Our experiment is focused on analyzing the impact of contrast change on the performance of the panoptic segmentation and increasing the model generalization on various image contrast settings using several approaches. This section describes the existing MaskDINO architecture, contrast enhancement, augmentation methods, and the research workflow.

4.1 MaskDino

MaskDINO [2] advances the DINO [18] architecture. DINO is a distinctive model of DETR [14], consisting of a backbone, encoder transformer, and decoder transformer. MaskDINO architecture is based on the DINO

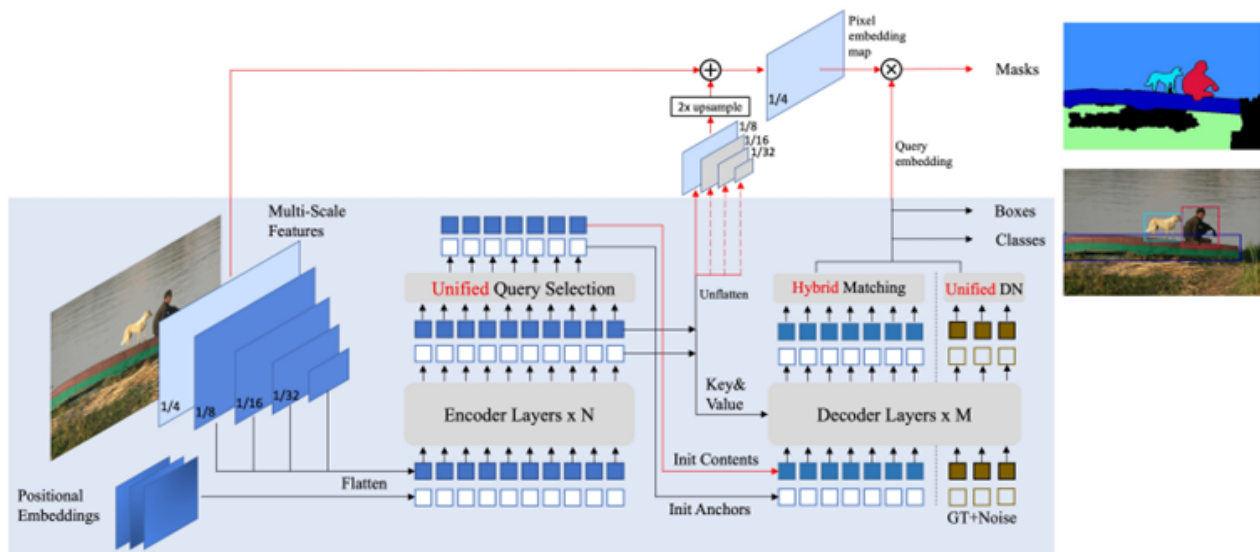


Figure 1: *MaskDINO* architecture

architecture (blue box) with slight modifications (red lines) for segmentation. In the decoder transformer, MaskDINO adds a mask branch for segmentation and includes several components in DINO for segmentation.


In the backbone, feature extraction produces feature maps with resolutions scaled at 1/4, 1/8, 1/16, and 1/32. In the segmentation branch, mask classification is performed for all segmentations using pixel embedding maps obtained from the backbone and encoder transformer features. In Figure 1, the pixel embedding map is generated by merging 1/4 resolution feature maps from the backbone and 1/8 resolution feature maps from the encoder transformer. Dot-product multiplication is performed on each content query embedding q_c from the decoder with the pixel embedding map to generate the output mask m .

$$m = q_c \otimes \mathcal{M}(\mathcal{T}(C_b) + \mathcal{F}(C_e)) \quad (1)$$

Where \mathcal{M} is the segmentation head, \mathcal{T} represents the convolutional layer to map its channel dimensions to hidden dimensions in the transformer, and \mathcal{F} represents a simple interpolation function to perform 2x upsampling on C_e . In MaskDINO, unified query selection is used to predict boxes and masks in the encoder and select the best one to start the query in the decoder. The selected mask and box serve as better initial references for the decoder. Then, unified denoising for masks accelerates convergence and improves performance, where the ground truth boxes containing noise and their labels are ingested in the decoder. The model is trained to reconstruct these ground truth boxes and masks. Hybrid matching is then performed to address the inconsistency in predictions of box-mask pairs generated from each head. This method performs bipartite matching between boxes and masks to encourage more accurate matching results.

For panoptic segmentation, box predictions for stuff categories are not required. Therefore, the box loss value and matching for stuff categories are removed. More specifically, the box prediction flow remains the same for stuff to find relevant areas and extract features with flexible attention. However, the box prediction loss value is not calculated and set to the average box prediction loss value for thing categories, which is done to accelerate model training.

Table 2: *COCO Panoptic JSON Nested Key-Values Pairs Details in 'annotations' value.*

Image	Key	Value	Example												
	segments_info	<p>The list of segmentation information of each of the objects in the image contains several key-value pairs</p> <table border="1"> <thead> <tr> <th>Key</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>id</td> <td>Segmentation ID in integer</td> </tr> <tr> <td>category_id</td> <td>Category id in integer based on categories list</td> </tr> <tr> <td>iscrowd</td> <td>Whether the object is crowded in a boolean (0 or 1)</td> </tr> <tr> <td>bbox</td> <td>Object bounding box coordinate in [x,y,width, height] format</td> </tr> <tr> <td>area</td> <td>Polygon area of the object in integer</td> </tr> </tbody> </table>	Key	Value	id	Segmentation ID in integer	category_id	Category id in integer based on categories list	iscrowd	Whether the object is crowded in a boolean (0 or 1)	bbox	Object bounding box coordinate in [x,y,width, height] format	area	Polygon area of the object in integer	<pre>"segments_info": [{ "id": 10462136, "category_id": 70, "iscrowd": 0, "bbox": [420, 295, 141, 147], "area": 16594 }, { "id": 12173263, "category_id": 81, "iscrowd": 0, "bbox": [149, 128, 268, 99], "area": 18368 }, { "id": 5008036, "category_id": 112, "iscrowd": 0, "bbox": [0, 72, 151, 408], "area": 32929 }, { "id": 3099247, "category_id": 118, "iscrowd": 0, "bbox": [90, 345, 531, 135], "area": 40029 }, { "id": 8622508, "category_id": 133, "iscrowd": 0, "bbox": [197, 0, 220, 115], "area": 22227 }, { "id": 10462914, "category_id": 195, "iscrowd": 0, "bbox": [463, 140, 41, 44], "area": 1536 }, { "id": 5002597, "category_id": 199, "iscrowd": 0, "bbox": [0, 0, 640, 420], "area": 116564 }]</pre>
Key	Value														
id	Segmentation ID in integer														
category_id	Category id in integer based on categories list														
iscrowd	Whether the object is crowded in a boolean (0 or 1)														
bbox	Object bounding box coordinate in [x,y,width, height] format														
area	Polygon area of the object in integer														

Tabel 2 COCO Panoptic JSON Annotation Format Key-Value Pairs Details (continued)

Image	Key	Value	Example
	File_name	File name of the image in string	"file_name": "000000061422.png"
	Image_id	Converted file name into an integer	"image_id": 61422

4.2 Modified Configuration For MaskDINO

We applied several modifications to enable the MaskDINO architecture to parse our custom dataset. This experiment used additional categories merged with the benchmark COCO Panoptic categories. Since there are additional new categories to the dataset, we modified the number of categories and the list of categories in the built-in metadata for the mask classification. This step is crucial to take before beginning training to ensure the correct categories assignment.

Throughout our experimentation with the custom dataset, we encountered a scenario where the trained model struggled to assign segmented images to any predefined category within the list. This issue surfaced as an error during the visualization of segmented images. To address this challenge, we created an additional function to assign unallocated segments to a list of unknown categories, enhancing the model's adaptability and effectively processing image or video inputs with diverse content and conditions crucial for the model testing on new unseen data.

4.3 Modified Preprocessing for MaskDINO

We experiment with various image contrast conditions, and several contrast settings and contrast enhancement methods are used for performance comparison. The input images are preprocessed using the validation set with several contrast value settings and contrast enhancement methods and evaluated one by one per setting and method. We used fixed contrast values of +50% and -50% for the contrast settings from the original image with a probability of 1. The contrast enhancement shows in figure 2 dan Figure 3

4.3.1 Histogram Equalization (HE)

Histogram Equalization is a digital image processing technique that enhances contrast in images achieved by spreading out the most frequently occurring intensity values, effectively stretching the intensity range of the image. This method typically enhances the global contrast in an image when the available data can be represented by values close to contrast. This method allows areas with lower local contrast to gain higher contrast [19]. Color histograms in an image represent the number of pixels in each color component. Histogram equalization cannot be applied separately to the image's red, green, and blue components because it would drastically change the image's color balance. However, suppose the image is first converted to another color space, such as the HSL/HSV color space. In that case, this algorithm can be applied to the luminance channel or value without altering the hue and saturation of the image.

4.3.2 Contrast Limited Adaptive Histogram Equalization

Contrast Limited Adaptive Histogram Equalization (CLAHE) is a variation of histogram equalization that limits the contrast value to prevent over-amplification of the contrast. CLAHE operates on the image segment result,

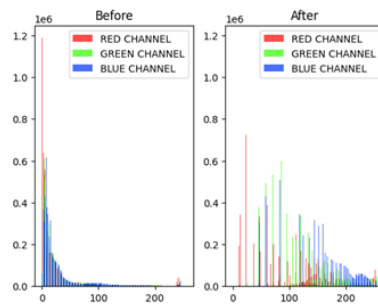


Figure 2: Comparison graph of RGB value distribution before (left side) and after (right side) Histogram equalization



Figure 3: Comparison of RGB images from the graph in Fig 2 before (left side) and after (right side) histogram equalization

called the grid, rather than the entire image [20]. Adjacent tiles are then combined using bilinear interpolation to remove artificial boundaries. CLAHE is defined by two parameters: `tileGridSize` and `Clip Limit (CL)`. `TileGridSize` assigns the number of tiles in the image row and column. `CL` assigns the contrast threshold [20].



Figure 4: Comparison graph of RGB value distribution before (left) and after (right) CLAHE with clip limit of 3 and tile grid size of 8 x 8

4.3.3 Gamma Correction

Gamma correction is a technique used to adjust the contrast and brightness of an image through a power law transformation. This transformation alters the gray levels of the input image to produce an enhanced output image. Denoting the gray levels of the input and output images as r and s , respectively, the transformation function can be expressed as:

$$s = T(r)$$



Figure 5: Comparison of RGB images from the graph in Fig 4 before (left) and after (right) CLAHE with clip limit of 3 and tile grid size of 8 x 8

In this transformation, the brightness enhancement is achieved by adjusting the r values to obtain corresponding s values. The formula gives the Power Law Transformation:

$$s = c * r^\gamma \quad (2)$$

Utilizes the parameter γ (gamma) to control the transformation. By varying γ , different results can be obtained, allowing for gamma correction to optimize the output image. For $\gamma < 1$, the darker regions of the original image become brighter, shifting the histogram towards the right. Conversely, for $\gamma > 1$, the opposite effect occurs, leading to adjustments in brightness and contrast accordingly.

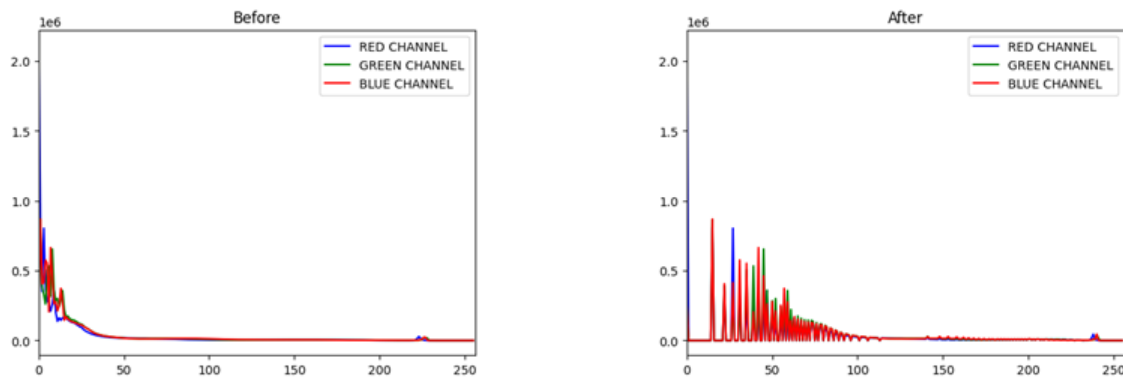


Figure 6: Comparison graph of RGB value distribution before (left side) and after (right side) gamma correction with $\gamma = 0.5$

4.4 Contrast Enhancement on Various Color Spaces

We ensure compatibility with various image representations, and contrast enhancement methods are tested in three color spaces: RGB, grayscale, and CMYK. We first converted the original images to the desired color space. Then, we applied histogram equalization to the converted images. The result examples are as follows in Table 3. Based on the table above, no results differ between the RGB and CMYK images after applying contrast enhancement. Therefore, the methods apply to both RGB and CMYK images. The methods also apply to grayscale images, resulting in relatively clear enhanced images, similar to the RGB and CMYK images.



Figure 7: Comparison of RGB images from the graph in Fig 7 before (left) and after (right) gamma correction with $\gamma = 0.5$

4.5 Modified Augmentation for MaskDINO

Image augmentation is a method to artificially create images based on the data images given through diverse processing or a combination of multiple processing methods. Image augmentation artificially expands the dataset to improve the model's generalization ability. The original augmentation methods are large-scale jittering and a size crop of 1024 x 1024. In this experiment, we use additional augmentation methods for two purposes:

1. to increase some of the category's representation and balance the category's representations.
2. to improve the model generalization toward various contrast settings.

For the first purpose, we applied horizontal flip, rotate by 30°, rotate by 15°, horizontal flip + rotate by 30°, horizontal flip + rotate by 15°, brightness decreased by 30 %, and brightness increased by 30 %. These will only be applied to the least represented categories and training-validation sets. For the second purpose, we applied fixed contrast values of +50 % and -50 % with a probability of 1. The augmentation mechanism for the second purpose is applied after the model is optimized without any augmentation.

Based on the table above, no results differ between the RGB and CMYK images after applying contrast enhancement. Therefore, the methods apply to both RGB and CMYK images. The methods also apply to grayscale images, resulting in relatively clear enhanced images, similar to the RGB and CMYK images.

4.6 Modified Augmentation for MaskDINO

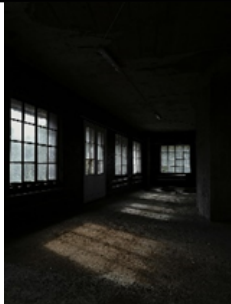










Image augmentation is a method to artificially create images based on the data images given through diverse processing or a combination of multiple processing methods. Image augmentation artificially expands the dataset to improve the model's generalization ability. The original augmentation methods are large-scale jittering and a size crop of 1024 x 1024. In this experiment, we use additional augmentation methods for two purposes:

1. to increase some of the category's representation and balance the category's representations.
2. to improve the model generalization toward various contrast settings.

For the first purpose, we applied horizontal flip, rotate by 30°, rotate by 15°, horizontal flip + rotate by 30°, horizontal flip + rotate by 15°, brightness decreased by 30 %, and brightness increased by 30 %. These will only be applied to the least represented categories and training-validation sets. For the second purpose, we

applied fixed contrast values of +50 % and -50 % with a probability of 1. The augmentation mechanism for the second purpose is applied after the model is optimized without any augmentation.

Table 3: *Contrast enhancement results on RGB, grayscale, and CMYK color spaces*

Step	RGB	Grayscale	CMYK
original image			
After HE			
After CLAHE			
After gamma correction			

5 Result and Discussion

We evaluated the performance of the MaskDINO model on the proposed dataset. The dataset comprises 7542 train images and 2756 validation images with nine additional categories (plug, plate, pan, box, toy, trash bin, wallet, fire extinguisher, and stove). The details of the data are as follows in Table 4 and Table 5.

Table 4: *Dataset distribution across training, validation, and test sets.*

Source	Train set	Validation set	Test set
COCO Panoptic 2017	2999	1021	-
Internet and phone recording	371	156	360
Augmentation	4172	1579	40
Total	7542	2756	400

Table 5: *Categories used in the dataset*

No	Super Category	Category	No	Super Category	Category
1	Person	Person	38	Raw Material	Cardboard
2	Electronic	Laptop	39		Paper
3		TV	40	Food-stuff	Food-other-merged
4		Mouse	41	Accessory	Handbag
5		Keyboard	42		Backpack
6		Cell phone	43		Wallet*
7		Plug*	44	Furniture-stuff	Chair
8	Appliance	Microwave	45		Dining table
9		Oven	46		Couch
10		Toaster	47		Bed
11		Sink	48		Toilet
12		Refrigerator	49		Table-merged
13	Kitchen	Spoon	50		Mirror-stuff
14		Bowl	51		Counter
15		Fork	52		Cabinet-merged
16		Knife	53		Door-stuff
17		Plate*	54		Light
18		Pan	55		Stairs
19		Stove*	56		Potted plant
20		Cup	57	Window	Window-blind
21		Bottle	58		Window-other
22	Indoor	Book	59	Ceiling	Ceiling-merged
23		Vase	60	Wall	Wall-brick
24		Trash bin*	61		Wall-wood
25		Fire extinguisher*	62		Wall-stone
26		Box*	63		Wall-tile
27		Toy*	64		Wall-other-merged
28		Clock	65	Floor	Floor-wood
29		Scissors	66		Floor-other-merged
30		Teddy bear			
31		Hair drier			
32		Toothbrush			
33	Textile	Blanket			
34		Curtain			
35		Pillow			

No	Super Category	Category	No	Super Category	Category
36		Towel			
37		Rug-merged			

5.1 Training Stage

Fine-tuning the MaskDINO model for this specific task involved several critical steps to adapt it to the varied and complex nature of indoor environments. Initially, the model was trained on the base dataset with standard hyperparameters, but the real improvement came through fine-tuning for contrast handling and further model optimization using the contrast-augmented dataset.

The primary focus during fine-tuning was to address the challenges posed by different contrast conditions in indoor images. To do this, the contrast values of the training data were augmented using fixed increases and decreases of 50 %, simulating lighting variations typically found in indoor spaces. This augmentation allowed the model to adapt to different lighting conditions, helping it recognize and segment objects more effectively, regardless of the contrast in the environment. This approach was a key differentiator from traditional contrast enhancement methods, which require preprocessing steps before model training. By applying contrast augmentation during the training phase, the model was able to learn to generalize across various contrast scenarios, resulting in better performance on unseen data with differing lighting conditions.

In addition to contrast handling, further fine-tuning was done by adjusting the training process to optimize for the specific indoor categories included in the dataset. The model's architecture was modified to account for additional categories beyond the typical COCO Panoptic dataset, such as appliances and household items. This required adjustments to the MaskDINO model's category configuration and the fine-tuning of its backbone (ResNet50) and learning rate settings. The learning rate was gradually decreased, and batch size was adjusted to accommodate the limitations of the hardware (V100 GPU with 16GB memory). This fine-tuning process helped improve the model's convergence and performance.

Augmentation strategies were also refined during this phase. For example, random horizontal flips, rotations, and brightness adjustments were applied to enhance the model's ability to handle spatial variations and lighting changes within indoor environments. Additionally, contrast augmentation, involving both contrast increase and decrease during the training phase, allowed the model to adapt to a wider range of lighting conditions without requiring additional contrast enhancement techniques like Histogram Equalization (HE) or CLAHE during the testing phase. This made the model more flexible and robust in various indoor settings.

Through these fine-tuning efforts, the model achieved significant improvements in segmentation and recognition quality, particularly in its ability to handle various contrast settings without overfitting to a specific contrast condition. The training procedure also involved experimenting with different contrast enhancement techniques (such as HE, CLAHE, and gamma correction), which further refined the model's performance, although contrast augmentation was found to provide the most significant impact on generalization.

Due to the resource limitation, the batch size was reduced from 16 to 2 for the hyperparameter settings. The model used the ResNet50 backbone and is trained for 45×4999 iterations. The initial learning rate was also reduced $\frac{1}{8}$ times from 1×10^{-4} to 1.25×10^{-5} . We also changed the lr scheduler mechanism, from dropping the lr by 0.1 times at the 11th epoch for the 12-epochs setting and the 20th epoch for the 24-epochs setting to dropping the lr by 0.1 times if the PQ did not improve in 1×4999 iterations.

5.2 Evaluation and Preprocessing Stage

The evaluation provides all the performance metrics information on the trained model, including PQ , PQ^{th} , PQ^{st} , RQ , RQ^{th} , RQ^{st} , SQ , SQ^{th} , and SQ^{st} . Once the model has reached the best PQ , it is going to be re-evaluated by preprocessing the image data with several contrast settings and contrast enhancement methods one by one, which are:

1. Contrast increase by 50 %
2. Contrast decreased by 50 %
3. Histogram equalization
4. Gamma correction with $\gamma = 0.5$

The evaluation of contrast settings is conducted to compare the performance of the model for each setting, and the evaluation of contrast enhancement methods is conducted to analyze the effectiveness of each method in improving the model performance.

5.3 Testing Stage

The testing aims to test the model segmentation ability on new unseen data. For the contrast experiment, each image will be tested in four different settings: original image, image increased by 50%, image decreased by 50%, and image using histogram equalization. The model will be tested in each setting, and each segmentation result will be analyzed for the segmentation existence and recognition precision. For the categories experiment, each image containing the category(s) will be tested and analyzed for segmentation and recognition ability and instance segmentation ability for things categories.

5.4 Augmentation Stage

Augmentation is applied to the training data to increase some of the category's representation and balance the categories' representations. We applied horizontal flip, rotate 30°, rotate 15°, horizontal flip + rotate 30°, horizontal flip + rotate 15°, brightness decreased by 30%, and brightness increased by 30%.

We proposed contrast augmentation to improve the model generalization if the contrast enhancement method is not enough for the improvement. We first applied a contrast decrease of 50% to the training and validation data until the model was optimized. Then, we applied a contrast increase of 50% and re-optimized the model.

5.5 Experimental results and analysis

The metrics used for the panoptic segmentation evaluation are as follows: PQ , PQ^{th} , PS^{st} , RQ , RQ^{th} , RQ^{st} , SQ , SQ^{th} , and SQ^{st} . SQ , or segmentation quality, is the metric to measure the model's ability to segment each object, where SQ^{th} is an average SQ for all categories, SQ^{th} for things categories, and SQ^{st} for stuff categories. RQ , or recognition quality, is the metric used to measure the model's ability to recognize the object after it is segmented. Panoptic Quality (PQ) is the multiplication of SQ and RQ . The formula for PQ computation is as follows:

$$PQ = \left(\frac{\sum_{(p,g) \in TP} \text{IoU}(p,g)}{|TP|} \right) \times \left(\frac{|TP|}{|TP| + \frac{1}{2}|FP| + \frac{1}{2}|FN|} \right)$$

where : The formula $\frac{\sum_{(p,g) \in TP} \text{IoU}(p,g)}{|TP|}$ is segmentation quality (SQ), and $\frac{|TP|}{|TP| + \frac{1}{2}|FP| + \frac{1}{2}|FN|}$ is recognition quality (RQ) with the requirement that a predicted segmentation and the ground truth segmentation can match (considered true positive (TP)) only if their intersection over union (IoU) is strictly greater than 0.5. True positive is in the context of SQ, every segmentation that is predicted true or matches with the ground truth, and in the context of RQ and category A, is every object that is recognized correctly as category A. False positive in the context of RQ and category A, is every object that is supposed to be recognized as other categories, but is falsely recognized as category A. False negative in the context of RQ and category A, is every object that is supposed to be recognized category A, but falsely recognized as another category.

Table 6: Comparative analysis of PQ, SQ, and RQ of proposed MaskDINO model from different contrast settings for 45 x 4999 iteration

Contrast setting	PQ (all)	PQ th	PQ st	SQ (all)	SQ th	SQ st	RQ (all)	RQ th	RQ st
No modification	47.2	50.2	43	79.5	83.8	73.4	55.5	57.9	52.3
Increase by 50%	41.0	44.4	36.4	77.8	82.2	71.8	48.9	51.5	45.2
Decrease by 50%	38.2	41.6	33.5	74.2	77.8	69.2	45.7	48.6	41.8

Based on the table 6, changes in contrast value affect the model performance. Either increased or decreased contrast decreased the value of the overall metric. We then compared several of the mentioned contrast enhancement methods in the table below.

Table 7: Comparative analysis of PQ, SQ, and RQ of proposed MaskDINO model from different contrast enhancement methods for 45 x 4999 iterations

Contrast setting	PQ (all)	PQ th	PQ st	SQ (all)	SQ th	SQ st	RQ (all)	RQ th	RQ st
No modification	47.2	50.2	43	79.5	83.8	73.4	55.5	57.9	52.3
HE	39.2	42.7	34.4	75.3	77.2	72.6	46.7	49.7	42.4
CLAHE	41.6	46.1	35.4	78.4	83.2	71.7	49.2	52.9	44
Gamma correction	45.2	48.3	40.9	81.6	87.4	73.4	52.7	54.8	49.7







Based on the table 7, either contrast enhancement is ineffective as it does not increase the overall PQ. However, without modification, gamma correction can surpass the overall SQ from the images. Gamma correction can improve images' overall quality by enhancing contrast and brightness. By adjusting the gamma value, which controls the relationship between pixel values and displayed brightness, gamma correction can effectively reveal details in the image's dark and light areas. This method can lead to more precise and more visually appealing images, making edges and objects within the image stand out more prominently. Even though gamma correction can achieve better SQ performance, it does not achieve a better RQ. Gamma correction can sometimes introduce unwanted color shifts in the image, resulting in unnatural or distorted colors, affecting the model's ability to recognize the objects within the image. This condition also goes for histogram equalization and CLAHE.

We then analyzed the panoptic segmentation result difference on the original image, decreased contrast and increased by 50% images, and applied histogram equalization image.

Table 8: Comparative analysis of the panoptic segmentation result of the proposed MaskDINO model from different contrast settings for 45 x 4999 iterations

Contrast Setting	Test Image	Panoptic Segmentation Result
-50%		

Table 8: Comparative analysis continued









Contrast Setting	Test Image	Panoptic Segmentation Result
0		
+50%		
Using HE		

Based on the results at tabel 8, several objects failed to be segmented, and some objects were segmented but failed to be recognized in different contrast settings. For instance, in the image with contrast decreased by 50%, the model failed to segment the lights, the window blinds, and the wooden floor. The model failed to segment the pillows and recognize the book in the histogram equalization image. The model also failed to segment curtains and recognize the window blind. We then proposed the second option, which is to apply contrast augmentation. We used the previous base model and continued the training with the same batch size and the last learning rate value. The training took 11 x 4999 additional iterations to reach the PQ value higher than the base model PQ value.

Table 9: Comparative analysis of PQ, SQ, and RQ of proposed MaskDINO model from different contrast enhancement methods for 45 x 4999 iterations

Contrast setting	PQ (all)	PQ th	PQ st	SQ (all)	SQ th	SQ st	RQ (all)	RQ th	RQ st
Base Model	47.2	50.2	43	79.5	83.8	73.4	55.5	57.9	52.3
HE	47.7	50.4	44	76	77.8	73.4	56.2	58.2	53.6

Table 10: Comparative analysis of the panoptic segmentation result of MaskDINO base model and contrast-augmentation-optimized from different contrast settings (CS)

CS	Base Model	Contrast-augmentation-optimized model
seg		
0		
+50%		
Using HE		

The details of whether objects in the image are segmented and recognized can be seen 11

Based on the comparison of object representation (Table 11 and Table 12), the contrast-augmentation-



Table 11: Object representation on base-model for Segmentation (Seg) and Recognition (Recog)

Object	contrast -50%		No contrast setting		contrast +50%		HE	
	Seg	Recog	Seg	Recog	Seg	Recog	Seg	Recog
Ceiling-merged	Yes	Yes	Yes	Yes	No	No	Yes	Yes
Wall-brick	No	No	Yes	Yes	Yes	Yes	Yes	Yes
Wall-other-merged	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Floor-wood	No	No	Yes	Yes	Yes	Yes	Yes	Yes
Window-blind	No	No	Yes	No	Yes	No	Yes	No
Window-other	No	No	Yes	Yes	Yes	Yes	No	No
Rug-merged	Yes	Yes	Yes	Yes	No	No	Yes	Yes
Counter	Yes	No	No	No	No	No	Yes	No
Table-merged	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Oven	Yes	No	Yes	No	Yes	No	Yes	No
Light	No	No	Yes	Yes	Yes	Yes	Yes	Yes
Couch	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Chair	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Clock	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
TV	No	No	No	No	No	No	No	No
Book	No	No	Yes	No	No	No	Yes	No
Pillow	Yes	Yes	Yes	Yes	No	No	No	No

optimized model produced slightly fewer segmentation and recognition mistakes than the base model. The contrast-augmentation-optimized model made one less segmentation mistake, which is 19 mistakes, than the base model, which is 20. The contrast-augmentation-optimized also made one less recognition mistake, ten mistakes, than the base mode, which made 11 mistakes.

However, due to the insignificant mistake differences, the contrast-augmented-optimized model is tested on images with various contrasts, including images that consist of dark and bright areas, rooms with similar color objects, and dim rooms.

Table 13: Segmentation (Seg) for Contrast Test Result

No	Test Image	Without Contrast Setting	With Contrast Setting
1	Original Image		 setting : Histogram equilization

Tabel 13 Segmentation (Seg) for Contrast Test Result (continued)

No	Test Image	Without Contrast Setting	With Contrast Setting
	Seg Result		
2	Original Image		 <p data-bbox="954 1256 1449 1294">setting : contrast decreased by 50 %</p>
	Seg Result		

Tabel 13 Segmentation (Seg) for Contrast Test Result (continued)

No	Test Image	Without Contrast Setting	With Contrast Setting
3	Original Image		 setting : contrast increased by 50 %
	Seg Result		
4	Original Image		 setting : contrast increased by 50 %
	Seg Result		

Tabel 13 Segmentation (Seg) for Contrast Test Result (continued)			
No	Test Image	Without Contrast Setting	With Contrast Setting
5	Original Image		 setting : contrast increased by 50 %
	Seg Result		
6	Original Image		 setting : histogram equalization
	Seg Result		

Based on Table 13, the segmentation result in both with and without contrast setting images produced similar results. In images 1 and 6, histogram equalization is applied to the image. Although it created a more refined

Table 12: *Object representation on contrast-augmentation-optimized model for Segmentation (Seg) and Recognition (Recog)*

Object	contrast -50%		No contrast setting		contrast +50%		HE	
	Seg	Recog	Seg	Recog	Seg	Recog	Seg	Recog
Ceiling-merged	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Wall-brick	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Wall-other-merged	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Floor-wood	No	No	Yes	Yes	No	No	Yes	No
Window-blind	No	No	Yes	No	Yes	No	Yes	No
Window-other	No	No	Yes	Yes	Yes	Yes	No	No
Rug-merged	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Counter	Yes	No	No	No	No	No	No	No
Table-merged	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Oven	Yes	No	Yes	No	Yes	No	Yes	No
Light	No	No	Yes	Yes	Yes	Yes	Yes	Yes
Couch	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Chair	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Clock	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
TV	No	No	No	No	No	No	No	No
Book	No	No	No	No	No	No	No	No
Pillow	No	No	Yes	Yes	No	No	No	No

contrast in image 1, the image gained noise from it, which can be seen from the wall segmentation that did not fully cover the wall area and the table that is not segmented. In image 6, histogram equalization made the objects' color change, which could affect the object's characteristic representation. It can be seen from the toilet segmentation that it did not fully cover the toilet area.

The model can perform panoptic segmentation without any additional contrast preprocessing. On the other hand, histogram equalization can not be applied universally to all types of images because it could produce noise and change the object characteristics, affecting the model's ability to represent the object. We then show several examples of the model panoptic segmentation qualitative result on several objects and compare the model quantitative performance to another method, which is the double-encoder network for RGB-D panoptic segmentation [21].

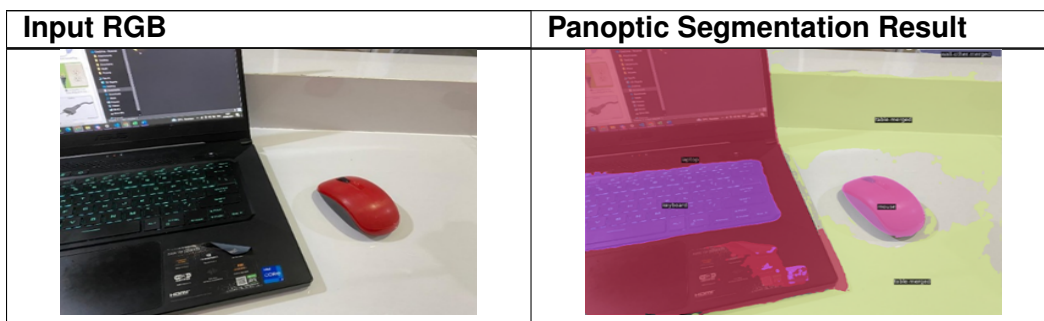
Table 14: *Qualitative results of the panoptic segmentation model*

Table 14: Qualitative results of the panoptic segmentation model (continued)









Input RGB	Panoptic Segmentation Result
	
	
	
	

Table 14: Qualitative results of the panoptic segmentation model (continued)



Input RGB	Panoptic Segmentation Result
	

Table 15: Comparison with other methods of panoptic segmentation in the indoor environment

Method	Dataset	PQ
Double-encoder network with ResidualExcite [21]	ScanNet	40.87
Double-encoder network with ResidualExcite [21]	HyperSim	38.67
MaskDino	COCO Panoptic, Open image primary dataset	47.70

In [21], ScanNet [22] and HyperSim [23] datasets are used for benchmarking. ScanNet contains real-world images organized in 1,513 scenes. HyperSim is a photorealistic synthetic dataset of indoor objects organized in 461 scenes. Our method uses a filtered COCO Panoptic 2017 dataset, additional open source images collected from the internet, and primary dataset which taken from phone recording for benchmarking. Based on the result above, our method still outperforms the previous method.

Table 16: Several Segmentation and Recognition mistakes in the proposed model










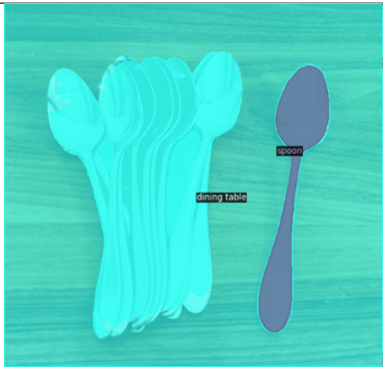


No	RGB Input	Panoptic Segmentation Result
1		
2		

Table 16: Several Segmentation and Recognition mistakes in the proposed model(continued)		
No	RGB Input	Panoptic Segmentation Result
3		
4		
5		
6		

However, we also found some limitations in our method. From Table 16 we perform in Figures number 1, 2, and 3, the model fails to recognize the objects correctly, where the socket is recognized as a microwave, the plates are recognized as bowls, and some parts of the forks are recognized as knives. This result could be because some object's characteristics are similar. For instance, the socket looks similar to a microwave button, the plates have a similar size, color, and dimension to bowls, and the forks are recognized as knives due to the light reflection on the fork handle that looks similar to a knife.

Our other limitation is found in Figures 4, 5, and 6. The model fails to segment some of the object instances. This result could be because other objects occlude some parts of the object. In Figure 4, some part of the

pink bowl is occluded by the yellow bowl, which causes an incomplete segmentation of the pink bowl. In Figure 5, the spoons are piled up onto each other and cause occlusion to one another. This condition can cause ambiguity in each spoon edge definition; hence, the model fails to segment them. Another possible factor is a low-resolution image. In Figure 6, the image dimensions are lower than those used in the dataset, which are 339 X 419 pixels. Low-resolution images tend to have fewer details and blurry object edge representation, thus making it difficult for the model to define the object segments.

6 Conclusion

This paper presented an approach to implementing panoptic segmentation for indoor environments and a method to mitigate various contrast challenges. Our method of using contrast augmentation can increase model generalization and reduce mistakes when handling various contrasts without additional contrast enhancement preprocessing. Our approach to using MaskDINO as the architecture and multiple indoor-related image sources as the dataset can outperform the previous method. Improved the model generalization on the various contrasts by re-optimizing it using a contrast-augmented dataset, resulting in impressive outcomes with a PQ score of 47.7%, a Recognition QuConality score of 56.2%, and a Segmentation Quality (SQ) score of 76%. These challenges highlight areas for future research and improvement. Specifically, addressing the model's performance on low-resolution images and refining occlusion handling could significantly enhance its robustness and applicability in real-world scenarios, indicating areas for future improvement. Future research could focus on enhancing model performance for low-resolution images, addressing occlusion challenges, implementing adaptive contrast adjustments, and expanding the model's scalability for real-world deployment, particularly in autonomous robotic systems operating in dynamic indoor environments.

7 Acknowledgements

This research was supported by the Research Grant Program for Faculty Members of the Department of Computer Science and Electronics, FMIPA, Universitas Gadjah Mada. We express our sincere gratitude to the program for the financial support and resources provided, which were essential to the successful completion of this study.

References

- [1] T. Y. Lin, M. Maire, S. Belongie, *et al.*, "Microsoft COCO: Common objects in context," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8693 LNCS, 2014, pp. 740–755. DOI: 10.1007/978-3-319-10602-1_48. arXiv: 1405.0312. [Online]. Available: http://link.springer.com/10.1007/978-3-319-10602-1_48.
- [2] F. Li, H. Zhang, H. Xu, *et al.*, "Mask DINO: Towards A Unified Transformer-based Framework for Object Detection and Segmentation," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2023-June, IEEE, Jun. 2023, pp. 3041–3050, ISBN: 9798350301298. DOI: 10.1109/CVPR52729.2023.00297. arXiv: 2206.02777. [Online]. Available: <https://ieeexplore.ieee.org/document/10204168/>.
- [3] Meta-AI, *The Leaderboard of Panoptic-Segmentation*, English, 2024. [Online]. Available: <https://paperswithcode.com/task/panoptic-segmentation> (visited on 01/08/2024).

- [4] Q. Yu, H. Wang, S. Qiao, *et al.*, “K-means mask transformer,” in *Computer Vision – ECCV 2022. Lecture Notes in Computer Science, vol 13689.*, S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, Eds., Cham: Springer Nature Switzerland, 2022, pp. 288–307, ISBN: 978-3-031-19818-2. DOI: https://doi.org/10.1007/978-3-031-19818-2_17.
- [5] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar, “Masked-attention Mask Transformer for Universal Image Segmentation,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2022-June, IEEE, Jun. 2022, pp. 1280–1289, ISBN: 9781665469463. DOI: 10.1109/CVPR52688.2022.00135. arXiv: 2112.01527. [Online]. Available: <https://ieeexplore.ieee.org/document/9878483/>.
- [6] Z. Li, W. Wang, E. Xie, *et al.*, “Panoptic segformer: Delving deeper into panoptic segmentation with transformers,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2022-June, IEEE, Jun. 2022, pp. 1270–1279, ISBN: 9781665469463. DOI: 10.1109/CVPR52688.2022.00134. arXiv: 2109.03814 [cs.CV]. [Online]. Available: <https://ieeexplore.ieee.org/document/9878498/>.
- [7] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, “Attention is all you need,” *Advances in Neural Information Processing Systems, NIPS’17*, vol. 2017-December, pp. 6000–6010, Jun. 2017, ISSN: 10495258. arXiv: 1706.03762. [Online]. Available: <http://arxiv.org/abs/1706.03762>.
- [8] A. Kirillov, R. Girshick, K. He, and P. Dollar, “Panoptic feature pyramid networks,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2019-June, IEEE, Jun. 2019, pp. 6392–6401, ISBN: 9781728132938. DOI: 10.1109/CVPR.2019.00656. arXiv: 1901.02446. [Online]. Available: <https://ieeexplore.ieee.org/document/8954091/>.
- [9] B. Cheng, M. D. Collins, Y. Zhu, *et al.*, “Panoptic-deeplab: A simple, strong, and fast baseline for bottom-up panoptic segmentation,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, Jun. 2020, pp. 12472–12482, ISBN: 978-1-7281-7168-5. DOI: 10.1109/CVPR42600.2020.01249. eprint: 1911.10194. [Online]. Available: <https://ieeexplore.ieee.org/document/9156495/>.
- [10] R. Mohan and A. Valada, “EfficientPS: Efficient Panoptic Segmentation,” *International Journal of Computer Vision*, vol. 129, no. 5, pp. 1551–1579, May 2021, ISSN: 15731405. DOI: 10.1007/s11263-021-01445-z. arXiv: 2004.02307. [Online]. Available: <https://link.springer.com/10.1007/s11263-021-01445-z%20http://arxiv.org/abs/2004.02307>.
- [11] J. Ren, C. Yu, Z. Cai, *et al.*, “REFINE: Prediction Fusion Network for Panoptic Segmentation,” *35th AAAI Conference on Artificial Intelligence, AAAI 2021*, vol. 3B, no. 3, pp. 2477–2485, May 2021, ISSN: 2159-5399. DOI: 10.1609/aaai.v35i3.16349. [Online]. Available: <https://ojs.aaai.org/index.php/AAAI/article/view/16349>.
- [12] Y. Li, H. Zhao, X. Qi, *et al.*, “Fully convolutional networks for panoptic segmentation with point-based supervision,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 4, pp. 4552–4568, 2023. DOI: 10.1109/TPAMI.2022.3200416.
- [13] W. Zhang, J. Pang, K. Chen, and C. C. Loy, “K-net: Towards unified image segmentation,” in *Proceedings of the 35th International Conference on Neural Information Processing Systems*, ser. NIPS ’21, Red Hook, NY, USA: Curran Associates Inc., 2024, ISBN: 9781713845393.
- [14] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, “End-to-End Object Detection with Transformers,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12346 LNCS, May 2020, pp. 213–229, ISBN: 9783030584511. DOI: 10.1007/978-3-030-58452-8_13. arXiv: 2005.12872. [Online]. Available: https://link.springer.com/10.1007/978-3-030-58452-8_13.

- [15] B. Cheng, A. G. Schwing, and A. Kirillov, “Per-Pixel Classification is Not All You Need for Semantic Segmentation,” in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. S. Liang, and J. W. Vaughan, Eds., vol. 22, Curran Associates, Inc., 2021, pp. 17 864–17 875, ISBN: 9781713845393. arXiv: 2107 . 06278. [Online]. Available: <https://proceedings.neurips.cc/paper/2021/file/950a4152c2b4aa3ad78bdd6b366cc179-Paper.pdf>.
- [16] J. Jain, J. Li, M. T. Chiu, A. Hassani, N. Orlov, and H. Shi, “OneFormer: One Transformer to Rule Universal Image Segmentation,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2023-June, IEEE, Jun. 2023, pp. 2989–2998, ISBN: 9798350301298. DOI: 10 . 1109/CVPR52729 . 2023 . 00292. arXiv: 2211 . 06220. [Online]. Available: <https://praeclarumjj3.github.io/oneformer/#BibTeX%20https://github.com/SHI-Labs/OneFormer%20https://ieeexplore.ieee.org/document/10203147/>.
- [17] Bergmark, *Intro to Image Processing in OpenCV with Python*, English, 2018. [Online]. Available: <https://medium.com/@pontus.bergmark/intro-to-image-processing-in-opencv-with-python-1c7f94af18a7> (visited on 05/30/2024).
- [18] H. Zhang, F. Li, S. Liu, *et al.*, “Dino: Detr With Improved Denoising Anchor Boxes for End-To-End Object Detection,” *11th International Conference on Learning Representations, ICLR 2023*, Mar. 2023. arXiv: 2203 . 03605. [Online]. Available: <http://arxiv.org/abs/2203.03605>.
- [19] S. Samsudin, *Introduction to Histogram Equalization for Digital Image Enhancement*, May 2021. [Online]. Available: <https://levelup.gitconnected.com/introduction-to-histogram-equalization-for-digital-image-enhancement-420696db9e43>.
- [20] Geeksforgeeks, *CLAHE Histogram Equalization – OpenCV*, May 2023. [Online]. Available: <https://www.geeksforgeeks.org/clahe-histogram-equalization-opencv/> (visited on 05/30/2024).
- [21] M. Sodano, F. Magistri, T. Guadagnino, J. Behley, and C. Stachniss, “Robust Double-Encoder Network for RGB-D Panoptic Segmentation,” in *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2023-May, IEEE, May 2023, pp. 4953–4959, ISBN: 9798350323658. DOI: 10 . 1109/ICRA48891 . 2023 . 10160315. arXiv: 2210 . 02834. [Online]. Available: <https://ieeexplore.ieee.org/document/10160315/>.
- [22] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, “ScanNet: Richly-annotated 3D reconstructions of indoor scenes,” in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-January, IEEE, Jul. 2017, pp. 2432–2443, ISBN: 9781538604571. DOI: 10 . 1109/CVPR . 2017 . 261. arXiv: 1702 . 04405. [Online]. Available: <https://ieeexplore.ieee.org/document/8099744/>.
- [23] M. Roberts, J. Ramapuram, A. Ranjan, *et al.*, “Hypersim: A Photorealistic Synthetic Dataset for Holistic Indoor Scene Understanding,” in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, IEEE, Oct. 2022, pp. 10 892–10 902, ISBN: 978-1-6654-2812-5. DOI: 10 . 1109/iccv48922 . 2021 . 01073. arXiv: 2011 . 02523. [Online]. Available: <https://ieeexplore.ieee.org/document/9711415/>.