Seamless Human-Background Integration via Refined Instance Segmentation and Deep Image Blending

Ashka Shah, Sharique Pervaiz, and Shivanshi Gupta

Abstract—Indoor human detection is essential for enhancing virtual reality (VR) systems by ensuring user safety and improving immersion. To address the lack of indoor pedestrian datasets, we propose an automated pipeline that transforms outdoor datasets into realistic indoor scenes, eliminating the need for manual mask creation. Our approach integrates Telea inpainting for seamless blending and style refinement using Deep Image Blending to produce high-quality augmented datasets. Experimental results demonstrate that models trained on our augmented datasets achieve superior detection performance compared to traditional blending techniques, with significant gains in mean Average Precision (mAP) and mean Average Recall (mAR). We have shown that in the absence of indoor datasets, our approach for augmenting outdoor datasets is a practical alternative. This work provides a scalable solution for dataset generation, supporting advancements in VR safety and interactivity.

1 Introduction

Accurate detection of moving individuals in indoor environments is crucial for enhancing virtual reality (VR) systems by preventing collisions, ensuring user safety, and fostering immersive interactions. However, the limited availability of publicly accessible indoor pedestrian datasets poses a challenge to advancing research in this domain. To address this, we propose a novel pipeline that transforms outdoor datasets into indoor equivalents, enabling the development of robust datasets tailored for indoor human detection.

While existing image blending techniques often rely on manually crafted masks, this approach can be time-consuming and inconsistent. Our method automates the data augmentation, streamlining the transformation from outdoor to indoor scenes. This automation facilitates the creation of comprehensive datasets essential for training effective indoor human detection models, with minimal reliance on manual input.

Key considerations in developing this pipeline include maintaining the realism and diversity of augmented indoor scenes and designing algorithms that accurately simulate indoor environments from outdoor data. By addressing these aspects, we successfully implemented a scalable and efficient data augmentation phase that supports the generation of high-quality datasets.

This work has significant implications for VR applications that rely on accurate indoor human detection. By addressing the dataset scarcity challenge, our method supports advancements in VR safety, usability, and immersion, contributing to the development of more responsive and interactive virtual environments.

2 Related Work

2.1 Laplacian Blending Technique

The Laplacian Blending Technique, first introduced by Burt and Adelson [1], decomposes images into multi-resolution Laplacian and Gaussian pyramids to facilitate seamless blending across image boundaries. This technique creates smooth transitions in the overlapping regions by combining spatial frequency information at different scales. Specifically, Gaussian pyramids are used to define blending masks with softened edges, while Laplacian pyramids enable the reconstruction of high-frequency details. This approach allows for the combination of images with minimal perceptual artifacts, effectively addressing issues like abrupt intensity changes or visible seams at the composite boundaries.

Step 1a. Build Laplacian pyramids LA and LB for images A and B respectively.
Step 1b. Build a Gaussian pyramid GR for the region image R.
Step 2. Form a combined pyramid LS from LA and LB using nodes of GR as weights. That is, for each l, i and j:

$$LS_l(i,j) = GR_l(i,j)LA_l(i,j) + (1 - GR_l(i,j))LB_l(i,j).$$

Step 3. Obtain the splined image S by expanding and summing the levels of LS.

Fig. 1. Laplacian Blending Methodology

Building on this foundational methodology, subsequent techniques have integrated advanced algorithms to address specific limitations. For instance, Poisson blending [2] incorporates gradient-domain consistency to ensure smooth transitions, even in scenarios involving varying intensities or textures. Similarly, GAN-based methods, such as GP-GAN [3], enhance visual realism by combining Laplacian blending principles with adversarial training to synthesize plausible high-frequency details.

Contemporary advancements leverage deep neural networks, incorporating style and content losses to achieve consistent textures and illumination in the blended regions. These methods adapt the underlying principles of Laplacian blending to modern applications like image harmonization, texture synthesis, and style transfer, demonstrating its enduring relevance and adaptability in computer vision tasks.

2.2 Poisson Blending Technique

Poisson blending, introduced by Pérez et al. [2], provides a robust approach to seamless image editing by leveraging the mathematical properties of the Poisson equation with Dirichlet boundary conditions. The technique formulates the blending task as a variational problem where the goal is to reconstruct the intensity values of a target region such that the gradient field aligns with a guidance vector field, typically derived from a source image. This ensures smooth transitions across the boundary of the blending region while preserving gradient consistency, effectively eliminating artifacts such as abrupt intensity changes or visible seams.

In practice, Poisson blending solves pixel intensities by minimizing the difference between the gradients of the blending region and the source image, subject to boundary constraints imposed by the target image. This approach enables natural blending even when the source and target regions differ significantly in texture or illumination. Unlike earlier methods such as multiresolution blending (Laplacian technique), which operate across multiple spatial frequency bands, Poisson blending achieves exact gradient field integration, resulting in precise control over the composite image's visual properties.

Extensions of Poisson blending have incorporated additional constraints or optimization strategies to enhance its applicability. For instance, gradient-domain [4] techniques generalize Poisson blending by allowing non-conservative guidance fields, enabling effects like texture enhancement, selective color adjustments, and transparent object insertion. More recently, neural network-based approaches [5] have integrated gradient-domain optimization with style and content losses, broadening the utility of the method in artistic rendering, image harmonization [6], and photorealistic editing. These advancements demonstrate the versatility of Poisson blending as a foundation for seamless image composition and its evolution into modern image processing pipelines.

2.3 Deep Image Blending Technique

Deep Image Blending [5] builds upon traditional techniques like Poisson Blending and extends them to address key limitations while incorporating advanced methodologies for improved results. Poisson Blending achieves seamless boundary transitions by enforcing gradient consistency between source and target images, but it struggles to adapt to target textures and often results in the over-blending of target colors into the source object. Additionally, its reliance on a closed-form solution makes it difficult to combine with other optimization objectives.

Deep Image Blending overcomes these challenges through a two-stage approach that integrates differentiable Poisson blending with style and content losses derived from deep neural networks. In the first stage, the algorithm enforces Poisson-based gradient-domain consistency to produce a seamless boundary while ensuring alignment between the source and target gradients. In the second stage, style and content losses are utilized to refine the texture and style of the blending region, achieving a balance between preserving the source content and adapting to the target's appearance.

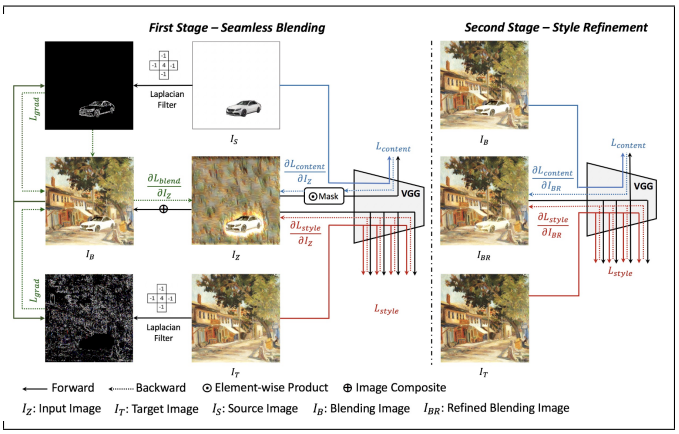

Fig. 2. Two-Step Image Blending Algorithm of Deep Image Blending [5]

Unlike supervised methods like GP-GAN [3], which require extensive paired training data, Deep Image Blending operates without training, making it versatile and generalizable across diverse image types, including real-world scenes and stylized paintings. Its differentiable formulation also allows for the inclusion of additional loss functions, such as histogram and total variation losses, to stabilize the blending process and enhance spatial smoothness.

Through this advanced framework, Deep Image Blending not only produces superior visual results compared to traditional Poisson Blending and hybrid techniques but also demonstrates robustness in handling complex blending tasks, including those involving inconsistent textures or illumination.

3 Proposed Method

Our goal is to seamlessly blend humans from outdoor to indoor scenes to generate a comprehensive data set that can later be used to train indoor human detection models. Blending techniques such as Laplacian blending [1] and Poisson blending [2] have been widely used to seamlessly merge cutouts to their background, yet they are limited in their application for our purposes. Laplacian blending ends up introducing blurriness and brightness artifacts onto the human cutout resulting in the loss of features. However, Poisson blending ends up introducing elements of the background into the human cutout. Seamless image blending techniques predominantly depend on manually crafted masks, an expensive process and labor intensive. This hinders the efficiency and scalability of the dataset creation process.

We propose an automated pipeline that can seamlessly blend humans from outdoor to indoor scenes in a computationally effective and realistic way.

3.1 Dataset Information

For pedestrian images, we use an outdoor pedestrian dataset [7][1], called the Penn-Fudan Database for Pedestrian Detection and Segmentation. This data set contains 170 images with 345 labeled pedestrians, of which 96 images are

1. The dataset can be accessed at https://www.cis.upenn.edu/~jshi/ped_html

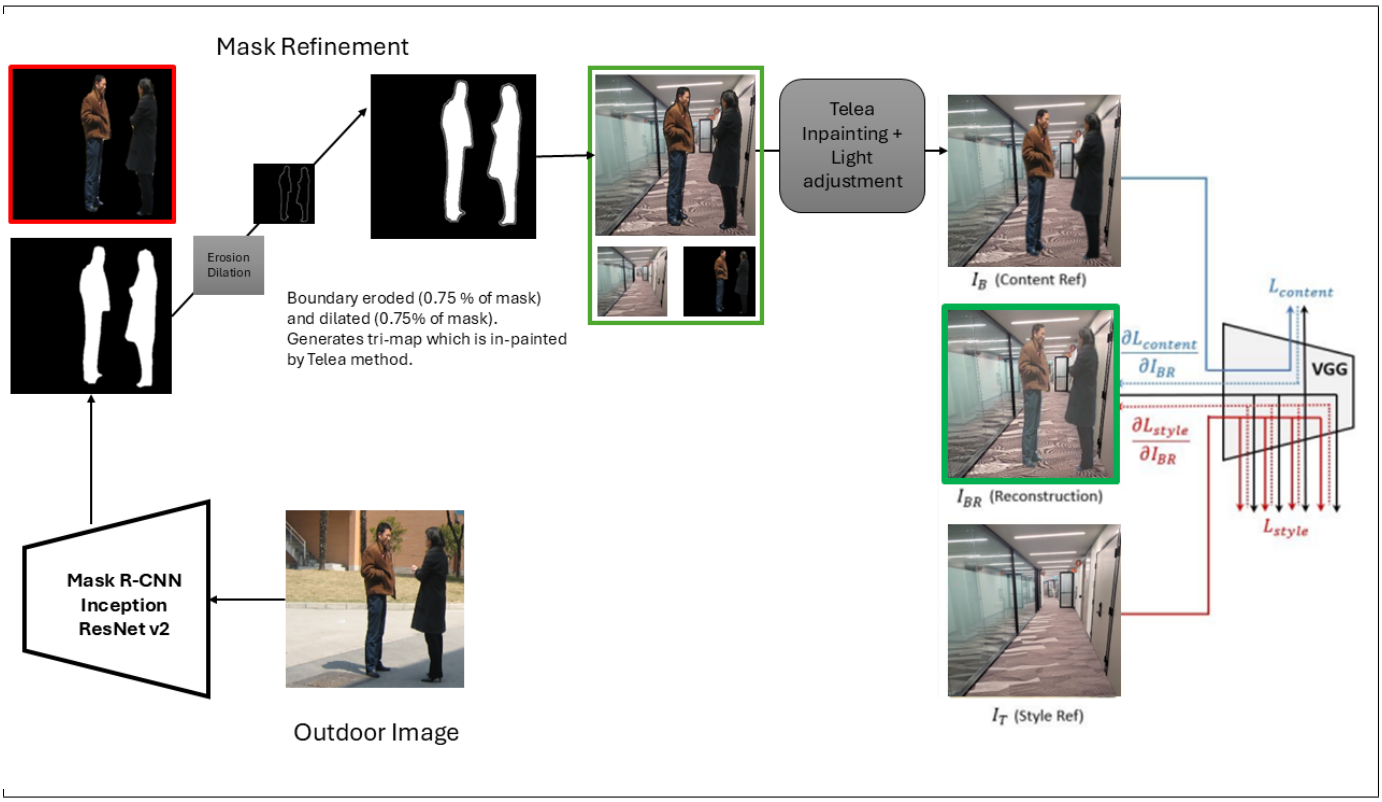

Fig. 3. A visual workflow of our proposed algorithm.

from the University of Pennsylvania and 74 are from Fudan University. For indoor scenes, we captured five images from different locations within the MScAC (Master of Science in Applied Computing) office, located on the 9th floor of 700 University Ave, Toronto.

3.2 The Technique

3.2.1 Obtaining Raw Human Masks

Each image from the Penn-Fudan dataset is processed through Mask R-CNN Inception ResNet V2 (Mask ResNet) [8][2], an image segmentation model trained on the COCO 2014 dataset [9] and published by TensorFlow on Kaggle. The Mask ResNet model returns masks for all object classes it can segment, but we focus solely on the "human" class. Any masks smaller than 9% of the image area (full image area is 400 x 400 pixels) are discarded. If more than two human masks are detected in an image, only the first two masks are kept, and the others are discarded. The bounding boxes are kept for each mask. Finally, all retained masks are combined into a single 400 x 400 pixel mask.

3.2.2 Generating a Tri-Map from Raw Mask

To refine the mask, we generate a tri-map. A **trimap** essentially consists of three regions: (1) the background region, which represents pixels from the background image; (2) the mid-boundary, an unknown region between background and foreground that will be estimated using information from both the background and the foreground pixels; and (3) the foreground region, which represents pixels from the human. See Figure 4.

Fig. 4. Sample trimap of a pedestrian. White pixels represent confirmed object, gray pixels represent mixture of object and background and black pixels represent confirmed background

First, we create an **erosion mask** by eroding the initial mask by 0.75% of its width. Then, we generate a **dilation mask** by dilating the initial mask by 0.75% of its width. Next, we subtract the erosion mask from the dilation mask to obtain a **trimap** mask, which contains a thin white boundary. The rationale for using the mask width to generate the erosion and dilation masks is to ensure the boundary is fine and proportional—neither too large nor too small.

3.2.3 Naive Light Adjustment

We also implement a naive light adjustment algorithm to adjust the lighting of the outdoor pedestrian image to the

2. The model can be accessed at https://tfhub.dev/tensorflow/mask_rcnn/inception_resnet_v2_1024x1024/1

lighting of the indoor background. First, we convert both the indoor and outdoor image to LAB color space [10]. LAB color space has 3 channels (1) L - light channel, represents light to dark from 0 to 100 (2) A- α channel that represents green to red transition from -127 to 128 (3) B- β channel that represents blue to yellow transition from -127 to 128 . We want to use the L channel since we are working with the lighting of the image.

Let L_{out} be the L-channel of the outdoor image and L_{in} be the L-channel of the indoor image. We then find the means of the L-channels of both images i.e. $\mu_{L_{out}}, \mu_{L_{in}}$ and the standard deviations of the L-channels of both images i.e. $\sigma_{L_{out}}, \sigma_{L_{in}}$. We then perform the following operation on L_{out}.

$$L_{out} = (\frac{L_{out} - \mu_{L_{out}}}{\sigma_{L_{out}}} \times \sigma_{L_{in}}) + \mu_{L_{in}}$$

The algorithm aims to normalize the L-channel of the outdoor image and then scale it by the lighting of the indoor image. We then pass this light adjusted outdoor pedestrian image along with its masks and the background image to the next step.

3.2.4 Cut-Paste and Telea Inpainting

The raw human mask is used to extract the human cutout from the light adjusted outdoor image, which is then pasted onto all five indoor backgrounds. This process is a simple cut-paste operation without depth correction, resulting in sharp intensity changes and edge imperfections at the boundary between the human and the background. To address these issues, we use the tri-map we generated earlier as a reference for correcting the intensity changes and edge artifacts from the cut-paste process. The coordinates of the mid-boundary region (as defined in Section 3.2.2) on the tri-map correspond to the unknown area on the blended image that will be inpainted using the **Telea inpainting technique** [11] by referring to the known pixels in the 5 pixel boundary of the unknown area.

The Telea inpainting technique uses neighbouring known pixels to fill in unknown regions, starting with the unknown pixels closest to the known pixels. In our case, the mid-boundary between the human and the indoor background represents the unknown region. When inpainting this mid-boundary, the unknown pixels adjacent to the background will be filled using the pixels from the indoor background, while those adjacent to the human will be filled using the pixels from the human. The unknown pixels that are equidistant from both the background and the human will be filled with information from pixels of the background and the cutout. This results in a more natural transition, minimizing sharp intensity changes and any background artifacts caused by an uneven mask. At this point, we have a blended image with a smooth transition. For the purposes of this pipeline, we use the OpenCV implementation of Telea Inpainting[3].

3.2.5 Style Refinement with Deep Image Blending Step 2

To seamlessly blend the human cutout into the indoor backgrounds, it is necessary to adjust the style of the blended image. Step 2 of the Deep Image Blending (DIB) algorithm focuses on transferring the style of the original background to the blended image's background while preserving the content of the cutout. This step primarily addresses contrast and lighting correction in the blended image. It uses the VGG16 model to optimize four loss functions: style loss, content loss and total variation (TV) loss (see Figure 5). For additional details on the exact loss functions, please refer to the Deep Image Blending paper [5] and codebase [4].

Algorithm 2 Second Stage - Style Refinement

Input: blending image I_B, target image I_T
max iteration T, loss weights $\lambda_{cont}, \lambda_{style}, \lambda_{tv}$
Given: a pretrained VGG network F
Output: refined blending image I_{BR}
$I_{BR} = copy(I_B)$
for $i \in [1:T]$ **do**
  $\mathcal{L}_{cont} = ContentLoss(I_{BR}, I_B, F)$
  $\mathcal{L}_{style} = StyleLoss(I_{BR}, I_T, F)$
  $\mathcal{L}_{tv} = TVLoss(I_{BR})$
  $L_{total} = \lambda_{grad} * L_{grad} + \lambda_{cont} * L_{cont} + \lambda_{style} * L_{style} + \lambda_{tv} * L_{tv}$
  $I_{BR} \leftarrow L\text{-}BFGS\_Solver(L_{total}, I_{BR})$
end

Fig. 5. Conceptual outline of Deep Image Blending Step 2. Taken from the original paper [5]

In this process, Step 1 of Deep Image Blending is replaced by the blended image generated through Telea inpainting from the previous step. We feed the resulting image into Step 2 of DIB and run the algorithm for 500 iterations (same hyperparameters as mentioned in the paper) to produce the final, seamlessly blended image. Lastly, we extract co-ordinates of the bounding boxes from the masks (an automated approach using `cv2.rectangle()` from OpenCV). This approach offers two key advantages over Step 1 of Deep Image Blending:

1. Step 1 of Deep Image Blending randomly initializes the background for the cutout and generates the entire background from scratch by optimizing on losses. This process is highly time-consuming and computationally expensive (1000 iterations take about 15 minutes). Additionally, it can introduce elements into the background that do not exist in the original image.
2. Since Step 1 of DIB optimizes using a Poisson loss (refer to the Deep Image Blending paper [5]), it often overlays color and styles from the background onto the human cutout. This can lead to significant loss of detail in the human features (see Experimental Results). By replacing Step 1 with the Telea inpainted image, we address the issue of introducing extraneous background elements and better preserve the human cutout, making the final blend more realistic.

3. Usage information can be found at https://docs.opencv.org/3.4/df/d3d/tutorial_py_inpainting.html

4. The implementation of this algorithm can be found at https://github.com/owenzlz/DeepImageBlending

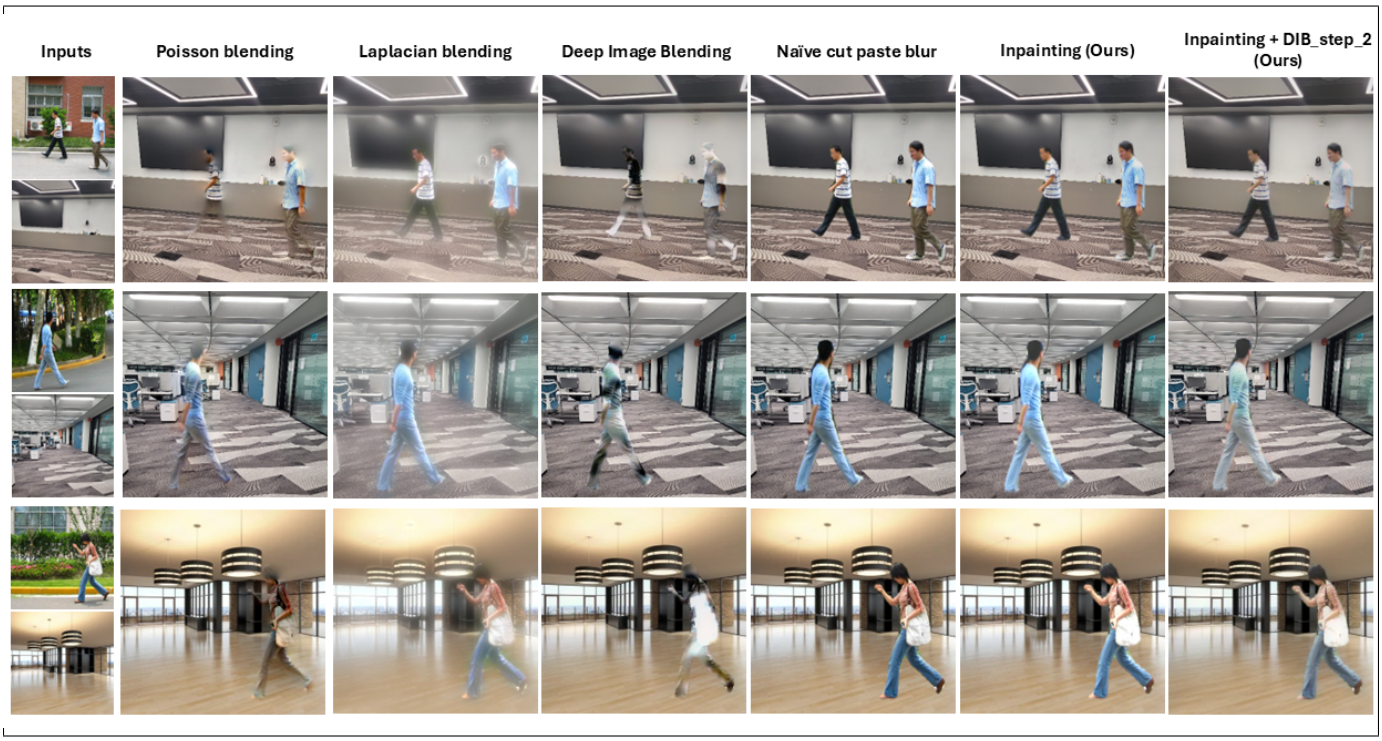

Fig. 6. This figure shows the comparison between the state-of-the-art image composite approaches and ours on outdoor pedestrians on indoor background images. Poisson Blending refers to **Poisson Image Editing [2]**. Laplacian refers to **Lapacian blending [1]**. Deep Image Blending (DIB) refers to **Deep Image Blending [5]** Naive cut paste blur refers to simply cutting outdoor image based on mask and pasting it onto background image

The images generated using these two methods make up the indoor human detection datasets.

4 Experimental

4.1 Experimental Setup

To curate a high-quality dataset, we filtered images from the Penn-Fudan Pedestrian dataset. Images with significant occlusions or where individuals were not appropriately scaled relative to the background were excluded. This filtering process resulted in a subset of 118 images. Each of these images was augmented using five distinct background images, leveraging our proposed trimap-based approach alongside several state-of-the-art blending techniques. These techniques included Poisson blending (using OpenCV's `cv2.seamlessClone()` implementation[5]), Laplacian blending, and simple cut-and-paste methods.

For the 118 filtered images, this augmentation process produced a total of 590 augmented images. Our approach utilized a trimap to ensure precise foreground-background segmentation, which was critical for achieving seamless blending. Each blending technique was applied uniformly across the dataset to maintain comparability. We evaluated our initial telea inpainting light adjustment augmentation seperately before applying deep image blending step 2 to evaluate the difference between them as well. The augmented samples were carefully inspected for quality, with examples shown in Figure 6 to highlight the differences in visual fidelity among the techniques.

5. For more details, refer to https://docs.opencv.org/4.x/df/da0/group__photo__clone.html

The resulting augmented dataset provides a diverse set of indoor scenes that closely mirror real-world conditions, making it well-suited for training object detection models for indoor pedestrian detection tasks.

4.2 Training Protocol

We trained separate instances of Faster R-CNN ResNet-50 [12] from scratch on each augmented dataset to assess the impact of different blending techniques. The datasets were split into 90% training and 10% validation subsets. For testing, we captured real-world images of individuals in our office environment with unseen backgrounds, ensuring that the test set was entirely independent of the training and validation data. This test set was specifically designed to simulate indoor pedestrian detection scenarios and evaluate the generalizability of the trained models.

Each model's performance was evaluated using metrics aligned with the COCO 2014 dataset [9], implemented via the pycocotools library. The evaluation included mean Average Precision (mAP) and mean Average Recall (mAR) to provide a comprehensive assessment of the models' detection capabilities. Notably, we excluded the dataset generated using deep image blending due to the subpar quality of its augmentations, as illustrated in Figure 6. Additionally, a baseline model was trained exclusively on the original outdoor dataset and evaluated on the same test set for comparison.

The training setup employed stochastic gradient descent (SGD) with a learning rate of 0.005, momentum of 0.9, and weight decay of 0.0005. The learning rate was decayed to 10% of its previous value every three epochs. The batch size

Models trained on different augmented datasets						
	Outdoors (Baseline)	Poisson	Laplacian	Simplae cut and paste	Inpainted (Ours)	Inpainted + DIB step 2 (Ours)
mAP IoU=0.50:0.95	0.478	0.534	0.581	0.592	0.627	**0.650**
mAP IoU=0.50	0.868	0.869	0.928	0.913	**0.937**	0.935
mAP IoU=0.75	0.523	0.621	0.670	0.763	**0.816**	0.793
mAR IoU=0.50:0.95	0.593	0.573	0.660	0.662	0.695	**0.702**

TABLE 1
Comparison of model trained on dataset with different augmentation techniques with baseline on coco evaluation metrics

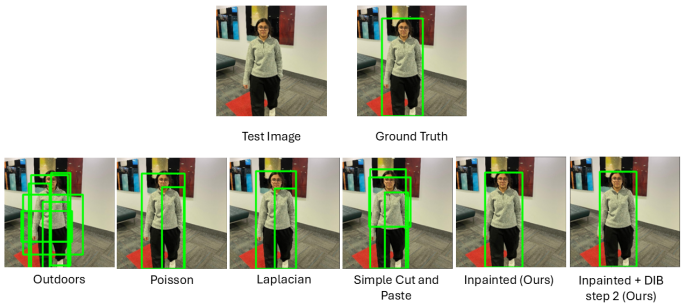

Fig. 7. Comparison of bounding boxes created by the model trained on different augmented datasets

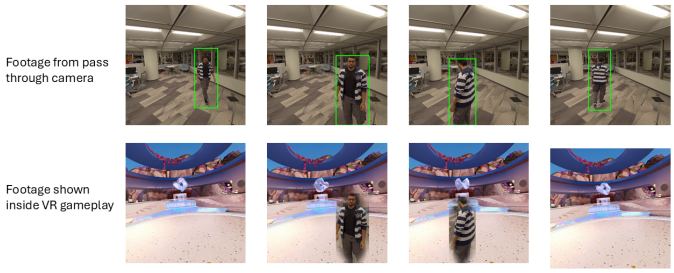

Fig. 8. POC of integrating model with MetaQuest pass through API showing person in passthrough VR when close to the user

was set to four images per batch. Training was conducted on an Nvidia RTX 4070 GPU, taking approximately 10 minutes per model, with an inference time of 42 ms per image. These parameters were chosen based on preliminary experiments to balance training efficiency and performance.

To ensure robust comparisons, each blending technique's dataset was trained using identical hyperparameters and data splits. This approach minimized confounding factors and allowed us to isolate the effect of the augmentation method on model performance.

4.3 Evaluation and Results

The performance metrics of the trained models are summarized in Table 1. Our proposed augmentation techniques consistently outperformed all other state-of-the-art blending methods. Specifically, the inpainting + DIB step 2 method achieved a mean Average Precision (mAP) of 65.0%, representing a 17.2% improvement over the baseline model trained on outdoor-only data. The mean Average Recall (mAR) was 70.2%, which is 10.9% higher than the baseline. These results highlight the superior quality of our augmentation technique in creating training data for indoor pedestrian detection.

Interestingly, even augmentations with visually unrealistic blending improved performance over the baseline, underscoring the value of augmenting pedestrian datasets for indoor environments. This observation suggests that augmentations introduce beneficial diversity into the dataset, helping the model generalize better to unseen indoor scenarios.

Among the techniques evaluated, simple cut-and-paste using the human mask outperformed Poisson and Laplacian blending. This finding suggests that for human detection in non-uniform backgrounds, Poisson and Laplacian blending may degrade overall augmentation quality. Our results indicate that maintaining clear object boundaries is more critical than achieving perfect visual realism for object detection tasks.

Visual examples in Figure 7 illustrate bounding boxes predicted by the models on test images. These examples show that models trained on datasets augmented using our proposed techniques generated more precise and accurate bounding boxes compared to those trained on other methods. The inpainting + DIB step 2 approach, in particular, produced bounding boxes with better localization and fewer false positives.

In summary, these results underscore the effectiveness of our augmentation strategy in enhancing object detection for indoor pedestrian scenarios. Our findings demonstrate the potential of leveraging targeted augmentation techniques to improve model performance in domain-specific applications, paving the way for further research in this area.

5 Conclusion

5.1 Summary

To conclude, we propose an automated data augmentation pipeline that seamlessly blend humans from an outdoor scene to an indoor scene allowing for the generation of comprehensive training datasets for indoor object detection models. In our case for human detection for VR, but can be extended to many other applications. We combine mask refinement via Telea inpainting with Step 2 of Deep Image Blending [5], replacing Step 1 of Deep Image Blending reducing the computational cost of the process. We also demonstrate that our indoor augmentation technique outperforms existing methods, such as Laplacian and Poisson blending, for indoor human detection. Additionally, we show that training an indoor object detection model using an outdoor pedestrian dataset is insufficient. In the absence of indoor datasets, our augmented indoor dataset yields optimal results. The code for this paper can be found at *https://github.com/A-Shah-ctrl/Seamless-Human-Background-Integration*

5.2 Limitations and Future Work

As a part of the project we aimed to integrate the indoor object detection system with the MetaQuest 3 VR headset. The idea was that when a human is detected in close proximity to the VR user, the human would pass through the VR scenes, alerting the user and helping to prevent collisions as shown in Figure 8. However, since Meta has not yet released the pass-through API for the MetaQuest 3, we were unable to complete the integration. We plan to proceed with this integration as soon as the API becomes available.

The dataset generation pipeline is relatively efficient compared to the full Deep Image Blending (DIB) algorithm; however, using Mask R-CNN Inception ResNet V2 [8] for image segmentation to obtain human masks presents a significant bottleneck. Step 2 of the DIB process requires approximately 500 iterations, which further slows down dataset generation. To improve efficiency, we could potentially remove DIB Step 2 from the pipeline, retaining only the mask refinement with Telea inpainting step. Performance results from the model trained on both Inpainting + DIB Step 2, and Inpainting alone, show very similar outcomes. Future work could focus on reducing segmentation time by exploring alternative image segmentation models, such as Grounded SAM [13] [6].

Acknowledgments

This paper was written as a part of the final project for CSC2529 Computational Imaging at the University of Toronto. The authors would like to thank Professor Aviad Lewis for his guidance and support throughout the project.

References

[1] P. J. Burt and E. H. Adelson, "A multiresolution spline with application to image mosaics," *ACM Transactions on Graphics (TOG)*, vol. 2, no. 4, pp. 217–236, 1983.

[2] P. Pérez, M. Gangnet, and A. Blake, "Poisson image editing," in *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, 2023, pp. 577–582.

[3] H. Wu, S. Zheng, J. Zhang, and K. Huang, "Gp-gan: Towards realistic high-resolution image blending," 2019. [Online]. Available: https://arxiv.org/abs/1703.07195

[4] R. Fattal, D. Lischinski, and M. Werman, "Gradient domain high dynamic range compression," in *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, 2023, pp. 671–678.

[5] L. Zhang, T. Wen, and J. Shi, "Deep image blending," in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 2020, pp. 231–240.

[6] Y.-H. Tsai, X. Shen, Z. Lin, K. Sunkavalli, X. Lu, and M.-H. Yang, "Deep image harmonization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 3789–3797.

[7] G. S. Liming Wang, Jianbo Shi and I. fan Shen, "Object detection combining recognition and segmentation," in *Asian conference on computer vision*, 2007.

[8] A. Yadav and E. Kumar, "Object detection on real-time video with fpn and modified mask rcnn based on inception-resnetv2," *Wireless Personal Communications*, pp. 1–26, 2024.

[9] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft coco: Common objects in context," 2015. [Online]. Available: https://arxiv.org/abs/1405.0312

[10] W. Collins, A. Hass, K. Jeffery, A. Martin, R. Medeiros, and S. Tomljanovic, "4.4 lab colour space and delta e measurements," Nov 2015. [Online]. Available: https://opentextbc.ca/graphicdesign/chapter/4-4-lab-colour-space-and-delta-e-measurements/

[11] A. Telea, "An image inpainting technique based on the fast marching method," *Journal of Graphics Tools*, no. 1, pp. 23–34, 2004.

[12] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *CoRR*, vol. abs/1506.01497, 2015. [Online]. Available: http://arxiv.org/abs/1506.01497

[13] T. Ren, S. Liu, A. Zeng, J. Lin, K. Li, H. Cao, J. Chen, X. Huang, Y. Chen, F. Yan, Z. Zeng, H. Zhang, F. Li, J. Yang, H. Li, Q. Jiang, and L. Zhang, "Grounded sam: Assembling open-world models for diverse visual tasks," 2024. [Online]. Available: https://arxiv.org/abs/2401.14159

6. More information on Grounded SAM can be found at https://github.com/IDEA-Research/Grounded-Segment-Anything