

# Predicting Depth Maps from Single RGB Images and Addressing Missing Information in Depth Estimation

Mohamad Mofeed Chaar<sup>ID</sup><sup>a</sup>, Jamal Raiyn<sup>ID</sup><sup>b</sup> and Galia Weidl<sup>ID</sup><sup>c</sup>

*Connected urban mobility, Faculty of Engineering,  
University of Applied Sciences, Aschaffenburg, Germany  
 {MohamadMofeed.Chaar, Jamal.Raiyn, Galia.Weidl}@th-ab.de*

**Keywords:** Depth image, image segmentation, Cityscape dataset, Autonomous Driving

**Abstract:** Depth imaging is a crucial area in Autonomous Driving Systems (ADS), as it plays a key role in detecting and measuring objects in the vehicle's surroundings. However, a significant challenge in this domain arises from missing information in Depth images, where certain points are not measurable due to gaps or inconsistencies in pixel data. Our research addresses two key tasks to overcome this challenge. First, we developed an algorithm using a multi-layered training approach to generate Depth images from a single RGB image. Second, we addressed the issue of missing information in Depth images by applying our algorithm to rectify these gaps, resulting in Depth images with complete and accurate data. We further tested our algorithm on the Cityscapes dataset and successfully resolved the missing information in its Depth images, demonstrating the effectiveness of our approach in real-world urban environments.

## 1 INTRODUCTION

Depth images are used to estimate the distance of objects at each pixel by employing a stereo camera. The Cityscapes dataset (Cordts et al., 2016; Cordts et al., 2015) Depth images are used to estimate the distance of objects at each pixel by employing a stereo camera. The Cityscapes dataset (Figure 1). However, a significant challenge in Depth imaging is the presence of missing information, which is a known issue in the Cityscapes dataset. Our work focuses on two key tasks:

- Generating Depth images from a single RGB image using a multi-level training approach based on an image segmentation algorithm.
- Enhancing the accuracy of Depth images by correcting missing information through our proposed methodology, achieving an accuracy rate of 90.19%.

In addition, our work enhanced the accuracy of Depth image generation from a single RGB image through image segmentation(Minaee et al., 2021; Yu et al., 2023).

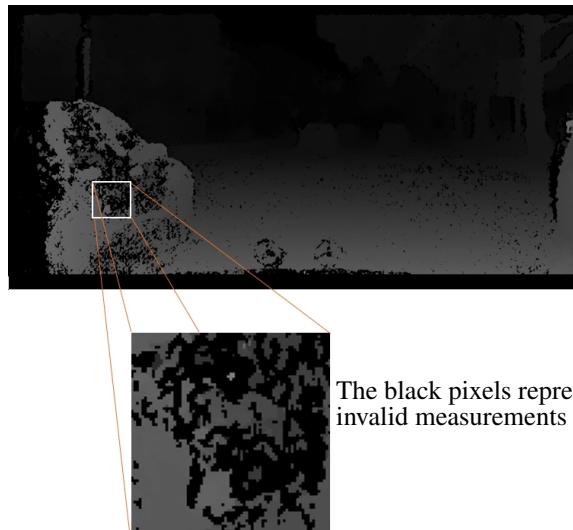


Figure 1: The Depth image from the Cityscape dataset contains black pixels, indicating missing information that affects the accuracy of the Depth image.

In a separate phase, we employed deep learning techniques to generate Depth images from the Cityscapes dataset, followed by applying our model to refine the data further. This refined dataset has been then used as a target to train a new model, continuing an iterative process (loop training) that has progressively improved the Cityscapes dataset and has resulted in

<sup>a</sup><sup>ID</sup> <https://orcid.org/0000-0001-9637-5832>

<sup>b</sup><sup>ID</sup> <https://orcid.org/0000-0002-8609-3935>

<sup>c</sup><sup>ID</sup> <https://orcid.org/0000-0002-6934-6347>

higher accuracy (Figure 2). In this study, we have applied the loop training process five times. After the first training iteration, the accuracy of generating Depth images reached 82.78%. In the last training loop, the accuracy increased to 90.19%, representing an improvement of approximately 7.41%.

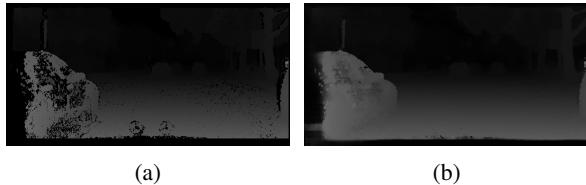


Figure 2: Image (a) shows an example from the Cityscapes dataset before applying our filtering technique to address missing pixels. Image (b) shows the result after applying our method, where the missing information has been successfully restored.

## 2 RELATED WORK

Monocular Depth estimation is the task of predicting a Depth map (distance from the camera) from a single RGB image. This is a challenging problem due to the lack of direct Depth cues (like stereo vision or motion) in a single image. Moreover, Many Depth estimation models trained on specific datasets tend to perform well on those datasets but fail when tested on new or unseen datasets due to differences in lighting conditions, camera characteristics, or environmental contexts. This is a significant issue because creating large-scale annotated Depth datasets is resource-intensive. The paper (Ranftl et al., 2020) mixed the data for better generalization, and the severe weather (Chaar et al., 2023; Chaar et al., 2024) has also been included in their data. The authors (Ma et al., 2019) introduced a method for Depth completion, converting sparse LiDAR data into dense Depth maps using RGB images without needing ground-truth Depth annotations. The approach leveraged self-supervised learning through photometric and geometric consistency losses, enabling the model to train effectively by using sparse LiDAR points and monocular images. It also incorporated stereo image pairs to enhance Depth accuracy. This framework achieved robust, high-quality Depth estimation, providing a practical solution for applications like autonomous driving, this methodology works well in specific scenes that are used in training, but it is limited when we generalize it in other data. The paper (Xian et al., 2018) approached predicting relative Depth from a single image using stereo data from the web for weak supervi-

sion. Instead of relying on costly ground-truth Depth annotations, the method leveraged geometric cues from stereo images (3600 images) to train a model that can generalize across diverse scenes. This scalable approach pushes the boundaries of monocular Depth estimation, making it effective for real-world applications without requiring extensive labeled data. The paper (Agarwal and Arora, 2023) introduced a monocular Depth prediction using a novel skip attention mechanism. The model leveraged attention layers focusing on important features across different network scales. By incorporating skip connections with attention, the model efficiently combines high-level semantic information with low-level spatial details, improving Depth prediction accuracy. The authors (Eigen et al., 2014) present an approach for predicting Depth maps from a single RGB image by utilizing a multi-scale deep network. The model processes images at different scales, capturing both global scene context and fine-grained details to improve Depth estimation accuracy. By combining features from multiple resolutions, the network is able to predict Depth for complex scenes better, handling variations in scale and structure more effectively. The paper (Zhang et al., 2023) introduced a novel approach for monocular Depth estimation using a multi-scale Laplacian pyramid to fuse Depth predictions at different scales. By capturing both global and local Depth information, the model refines Depth estimates through residual learning at each pyramid level.

Previous studies primarily focused on generating Depth images by utilizing Depth images as the target output. In contrast, our methodology has introduced improvements to the datasets by addressing and correcting missing pixels, allowing us to use the refined Depth images as target outputs. This approach resulted in a significant increase in accuracy, achieving a performance of 90.19% as we will see later.

## 3 Methodology

### 3.1 Image Segmentation

Image segmentation is the process of partitioning an image into multiple segments or regions for each pixel, with the goal of simplifying the representation of an image to make it more meaningful and easier to analyze. One of the most effective architectures for image segmentation tasks, especially in medical imaging, is the U-Net architecture (Ronneberger et al., 2015), which operates the encoder and decoder layers to deduce the segmentation for each pixel. In

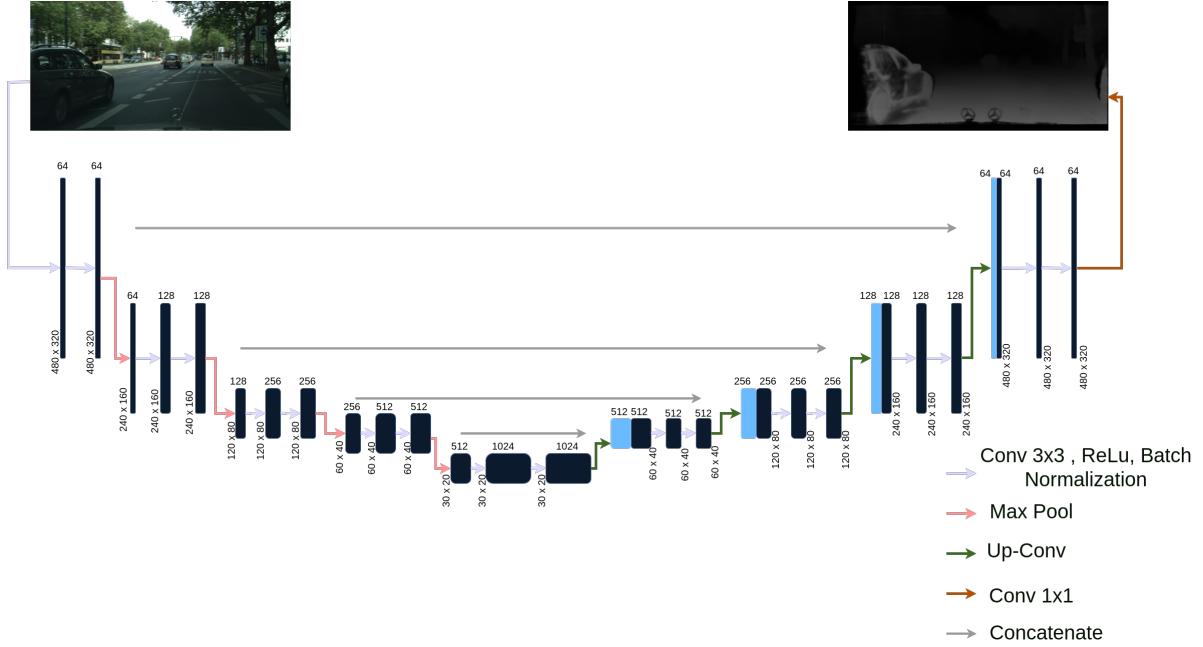


Figure 3: Creator of the U-Net algorithm, which was implemented in our work.

In practice, U-Net can be trained using annotated images where each pixel is labeled as belonging to a specific class or object. The loss function that we used in this work is the Mean Square Error Loss function (MSELoss). In our work, we used U-Net to train on the Cityscapes dataset, with the Depth image as the target mask. Both input and output images have a resolution of 480x320 pixels (Figure 3).

The stereo camera provides two images (left and right) for each depth image, but since our goal is to generate depth from a single image, we selected only the left image for training and omitted the right image.

## 3.2 Metrics

### 3.2.1 Accuracy

The accuracy in this work is evaluated based on the Absolute Error between the predicted pixel values and the target values of the depth image, calculated using the following formula:

$$\text{Absolute Error pixels} = \sum_{P \in \text{Pixels}} |P - \hat{P}| \quad (1)$$

Where  $P$  represents the actual pixel value, and  $\hat{P}$  denotes the predicted pixel value, the percentage of accuracy is calculated by dividing the absolute error of the pixels by the sum of all pixel values in the image, as expressed in the following formula:

$$\text{Accuracy} = \frac{\text{Absolute Error pixels}}{\sum_{P \in \text{Pixels}} P} * 100 \quad (2)$$

### 3.2.2 Corrected Pixels

The primary objective of this work is to correct missing information in depth images, represented as black pixels. To establish a criterion for addressing this issue, we calculated the average number of black pixels per image (Appendix A), which was found to be 1,206,898. To validate our approach, we calculated the average number of corrected black pixels per iteration, using this as a performance criterion, as expressed in the following formula which calculate the percentage of corrected pixels:

$$\text{Corrected Pixels} = \frac{\text{Average of corrected pixels}}{\text{Average black pixels}} * 100 \quad (3)$$

## 3.3 Depth image

Typically, generating a Depth image dataset involves using a stereo camera(Zhou, 2024) setup, where two cameras capture images from slightly different angles. The Depth information is obtained by calculating the disparity between corresponding points in the two images.

Estimate Depth of pixels is calculated by the following equation(OpenCv, 2024):

$$Z = Bf / \text{disparity} \quad (4)$$

Where  $B$  (baseline) is the distance between two cameras where it is 22 cm in the Cityscapes dataset (Cordts et al., 2016),  $Z$  is the Depth of the pixel, and  $f$  is the focal length.

In cityscape datasets, *disparity* calculated by the following (Team, 2021):

$$\text{disparity} = (\text{float}(p) - 1.) / 256 \quad (5)$$

Where  $p$  is the value of a pixel in the Depth image, and it is between [0,126]; furthermore,  $p = 0$  when the measurement is invalid (Figure 4) (Kaehler and Bradski, 2016).

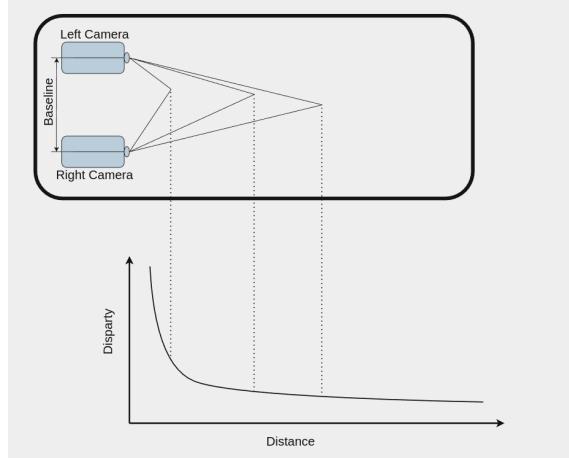


Figure 4: The disparity value can be used to calculate distance using Equation 4

### 3.4 Cityscape Datasets

The Cityscapes dataset (Cordts et al., 2016; Cordts et al., 2015; Downloads, 2016) is a large-scale dataset created for urban scene understanding, particularly tailored for research in semantic segmentation, instance segmentation, and depth prediction in complex city environments. This dataset is widely used in computer vision research due to its high-quality annotations and realistic street scenes captured from various German cities. The depth images in the Cityscapes dataset were generated using stereo camera sensors mounted on a vehicle as it drove through various European cities. The dataset is organized into three folders: train, test, and validation, containing a total of approximately 150,000 images with size (2048x1024) pixels. In addition to the depth images, corresponding left and right camera images are provided, enabling comprehensive analysis and training for depth estimation and other vision-based tasks. As shown in Figure 5, the depth image contains black points, representing invalid measurement areas. In

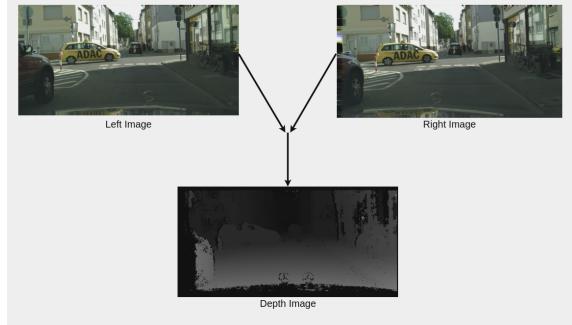


Figure 5: The image was captured from the Cityscapes dataset in Frankfurt. Using the left and right stereo images, a depth image can be generated.

the Cityscapes dataset, each image has an average of 1,206,898 black pixels (Appendix A), accounting for 57.5% of the image, indicating missing information, which our work is concerned with solving this problem.

### 3.5 Iterative training

The missing information impacts the accuracy of the training data, reducing accuracy when this data is used as a target mask in image segmentation algorithms, resulting in an accuracy of 82.78% (Figure 6)

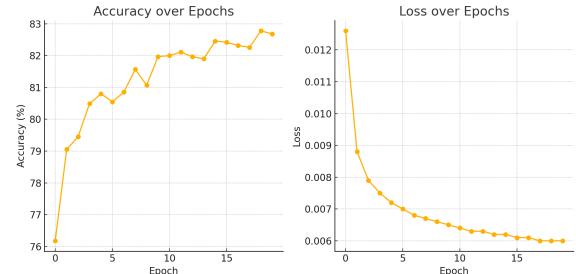


Figure 6: the graphs showing how accuracy and loss change over the steps. The left graph displays the accuracy trend, while the right graph shows the loss trend across each step; the highest accuracy is 82.78%.

The image segmentation results after training the data give us a result without missed information which means a smoother image but with an accuracy of 82.78% for the pixels that have been filled in Depth image (Figure 7).

We trained our model for image segmentation to enhance the Depth of images and address the issue of black pixels. After predicting each image, we examined every pixel in the Depth image from the Cityscape dataset. The zero-value pixels are replaced with the corresponding pixel values predicted by our

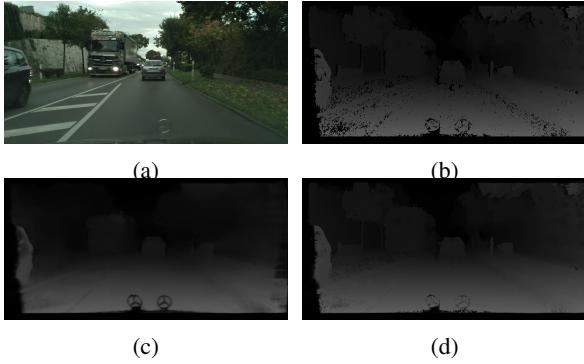


Figure 7: Figure (a) presents an example from the Cityscapes dataset captured in Lindau city. Figure (b) depicts a Depth map generated using the stereo camera. Figure (c) illustrates the predicted depth map produced by our initial model training. Finally, Figure (d) demonstrates an enhanced version of the depth map generated by the stereo camera (Figure b), achieved through our proposed method, as reflected in the improvements shown in Figure (c).

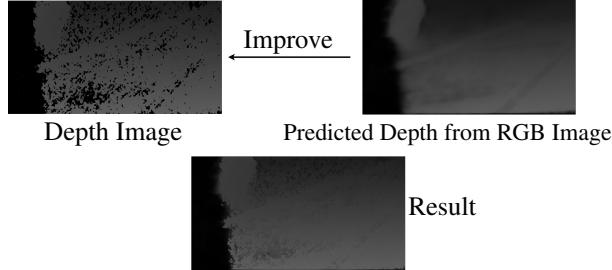


Figure 8: The Depth image from the Cityscape dataset contains black pixels, indicating missing information that affects the accuracy of the Depth image.

model for the same location, effectively filling in the missing Depth information (Algorithm 1)(Figure 8).

```

Data: Depth image
Result: Fix missing information in Depth
    image
    initialization;
    predict depth image using our model;
while There is Pixel with value zero do
    | Replace it with the value of the predicted
    | image in the same position;
end

```

Algorithm 1: How to Fix black pixels in Depth image

The results shown in Figures 7 and 8 illustrate our initial training (first iteration). At this stage, we enhanced the Depth images, achieving complete images without missing information, which we filled using our predictions with an accuracy of 82.78%. These improved images can now be utilized to train our data further and serve as masks for image segmentation in the next training iteration. This approach increases

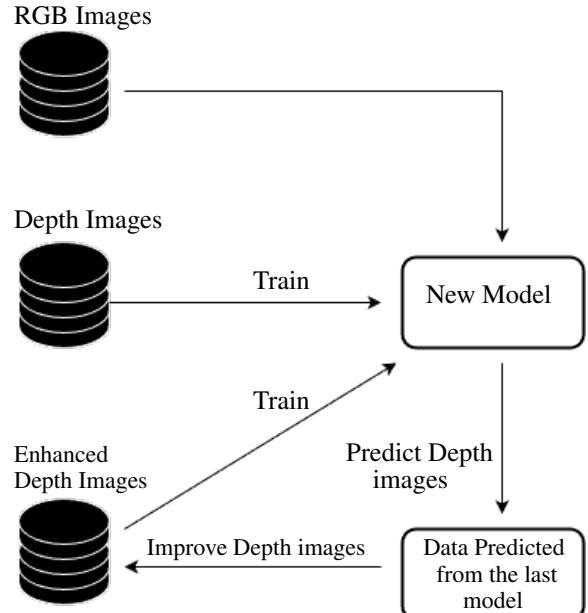


Figure 9: This diagram illustrates the iterative training process, where we train our datasets to predict depth images. We then enhance the depth images in the Cityscape dataset and retrain using the enhanced images. With each iteration, we observe an increase in accuracy. In this work, we repeated the training process five times.

accuracy, as the masks used are free from missing information (Algorithm 2) (Figure 9).

```

Data: Depth image
Result: Depth image without missed
    information
while Iteration do
    | Train a new model using Depth images as
    | a mask;
    | Improve Depth image using our model
    | (Algorithm 1);
    | Update Depth images by improved Depth
    | images;
end

```

Algorithm 2: In iterative training, we process the datasets using the latest depth images, which we have enhanced.

When using enhanced depth images, the accuracy of training the dataset improved with each iteration. During the first training iteration, an accuracy of 82.78% was achieved. After enhancing the datasets, the accuracy increased to 88.16%. Based on this improvement, we decided to iterate the training process five times, ultimately achieving an accuracy of 90.19%.

Table 1: The accuracy and the percentage of corrected pixels improved significantly through our iterative learning process. As observed, the training accuracy of our model increased steadily, and the percentage of corrected black pixels also showed a consistent rise.

Iterative	Accuracy <sup>1</sup>	Corrected Pixels <sup>2</sup>
1	82.78%	58.3%
2	88.16%	65.8%
3	89.11%	70.5%
4	89.83%	72.5%
5	90.19%	73.3%

<sup>1</sup>According to our formula at Equation 2

<sup>2</sup>According to our formula at Equation 3

### 3.6 Correct Missed Information Using U-Net

According to our previous work, we improved depth images by filling in the missing information with highly accurate predictions during the final iteration of training (90.19%). However, the correction process is time-consuming and impractical, as it requires predicting each image and checking every pixel to fill in the missing information (806 ms for each frame) for our computer properties, see Appendix B. Fortunately, we already have depth images and their corrected versions from our earlier work. This allows us to train a new image segmentation model, using the depth images with missing information as input and the corrected images as masks. The model can then predict new images with corrected missing information using AI, where we have achieved an accuracy of 96.44% (Figure 10d) which it costs (19 ms for each frame).

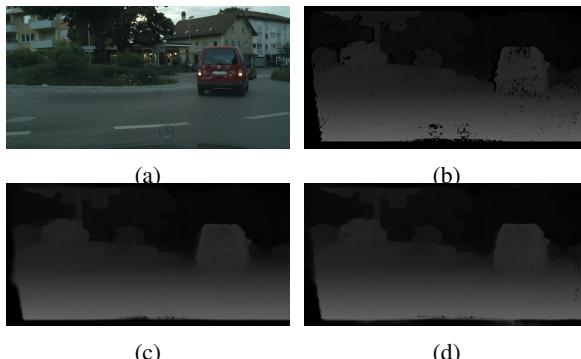


Figure 10: Figure (a) presents an example from the Cityscapes dataset captured in Lindau city. Figure (b) depicts a Depth map generated using the stereo camera. Figure (c) illustrates our model to enhance the missed information from the depth image using the U-Net algorithm (the input is an image (b)). Finally, Figure (d) demonstrates an enhanced version of the depth map generated by the stereo camera (Figure b), achieved through our proposed method, as reflected in the improvements shown in Figure (d).

## 4 RESULTS AND DISCUSSION

Building upon our previous approach, we successfully accomplished multiple tasks in this work, as outlined below:

- Generating depth images from a single RGB image using an iterative training approach, we steadily improved the accuracy of depth predictions with each iteration. The iterative process demonstrated consistent progress, with accuracy increasing at each step, showcasing the effectiveness of the refinement and optimization strategies applied during training.
- Improving Depth Image Accuracy with U-Net and Self-Supervised Learning: Leveraging enhanced depth maps as masks to correct missing information, achieving 96.44% accuracy.
- 

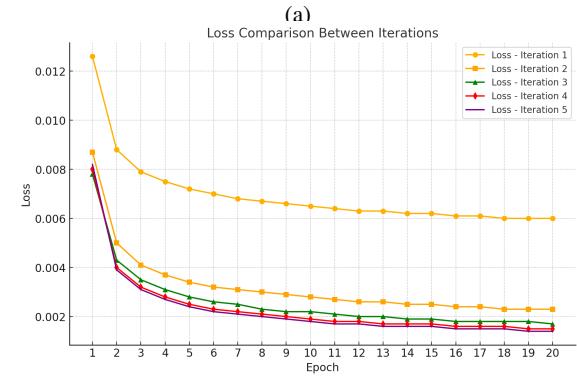
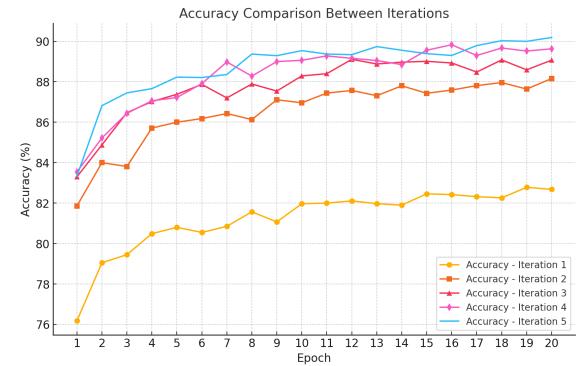


Figure 11: Figure a: Accuracy Trends Across Epochs for Multiple Iterations This figure illustrates the progression of accuracy across epochs for five iterations, highlighting improvements and comparisons between each iteration. Figure b: Loss Trends Across Epochs for Multiple Iterations This figure depicts the reduction of loss across epochs for five iterations, demonstrating the convergence behavior and performance stability over time.

## 5 CONCLUSIONS

This study demonstrates a comprehensive approach to addressing the challenge of missing information in Depth images, a critical aspect of Autonomous Driving Systems (ADS). By employing iterative training combined with deep learning techniques, we successfully generated and refined Depth images from single RGB inputs, achieving a notable accuracy improvement from 82.78% to 90.19% over five iterations. The iterative refinement process not only enhanced the accuracy but also significantly reduced the percentage of missing pixels in Depth images.

Additionally, leveraging the U-Net algorithm for image segmentation allowed us to automate and accelerate the correction process, further improving the prediction accuracy to 96.44%. These advancements were validated using the Cityscapes dataset, which served as an effective benchmark for urban scene understanding in autonomous driving applications. Our methodology demonstrated robust performance in filling missing information, as evidenced by substantial improvements in corrected pixel percentages and accuracy metrics.

This work lays a strong foundation for future research aimed at enhancing Depth image generation and refinement techniques. The iterative training approach and segmentation-based corrections can be extended to other datasets and use cases, such as 3D reconstruction, robotics, and other computer vision applications, where accurate depth information is paramount. Future directions may include optimizing the computational efficiency of the model and exploring multi-modal input strategies to further improve depth prediction performance in real-time scenarios.

## REFERENCES

- Agarwal, A. and Arora, C. (2023). Attention attention everywhere: Monocular depth prediction with skip attention. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5861–5870.
- Chaar, M., Weidl, G., and Raiyn, J. (2023). Analyse the effect of fog on the perception. *EU Science Hub*, page 329.
- Chaar, M. M., Raiyn, J., and Weidl, G. (2024). Improving the perception of objects under foggy conditions in the surrounding environment. *Research square*.
- Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., and Schiele, B. (2016). The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223.
- Cordts, M., Omran, M., Ramos, S., Scharwächter, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., and Schiele, B. (2015). The cityscapes dataset. In *CVPR Workshop on The Future of Datasets in Vision*.
- Downloads, C. D. (2016). Cityscapes dataset downloads. <https://www.cityscapes-dataset.com/downloads/>. Accessed: February 5, 2025.
- Eigen, D., Puhrsch, C., and Fergus, R. (2014). Depth map prediction from a single image using a multi-scale deep network. *Advances in neural information processing systems*, 27.
- Kaehler, A. and Bradski, G. (2016). *Learning OpenCV 3: computer vision in C++ with the OpenCV library*. ” O'Reilly Media, Inc.”.
- Ma, F., Cavalheiro, G. V., and Karaman, S. (2019). Self-supervised sparse-to-dense: Self-supervised depth completion from lidar and monocular camera. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 3288–3295. IEEE.
- Minaee, S., Boykov, Y., Porikli, F., Plaza, A., Kehtarnavaz, N., and Terzopoulos, D. (2021). Image segmentation using deep learning: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(7):3523–3542.
- OpenCv (2024). Depth map from stereo images. Last accessed 18 November 2024.
- Ranftl, R., Lasinger, K., Hafner, D., Schindler, K., and Koltun, V. (2020). Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE transactions on pattern analysis and machine intelligence*, 44(3):1623–1637.
- Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5–9, 2015, proceedings, part III 18*, pages 234–241. Springer.
- Team, C. (2021). cityscapesscripts. Last accessed 17 November 2024.
- Xian, K., Shen, C., Cao, Z., Lu, H., Xiao, Y., Li, R., and Luo, Z. (2018). Monocular relative depth perception with web stereo data supervision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 311–320.
- Yu, Y., Wang, C., Fu, Q., Kou, R., Huang, F., Yang, B., Yang, T., and Gao, M. (2023). Techniques and challenges of image segmentation: A review. *Electronics*, 12(5):1199.
- Zhang, A., Ma, Y., Liu, J., and Sun, J. (2023). Promoting monocular depth estimation by multi-scale residual laplacian pyramid fusion. *IEEE Signal Processing Letters*, 30:205–209.
- Zhou, R. (2024). Scalable multi-view stereo camera array for real-time image capture and 3d display in real-world applications. *Mathematical Modeling and Algorithm Application*, 2(2):43–48.

## Appendix A

We calculate the average number of black pixels across all Depth images in the Cityscapes dataset using the following formula:

$$\text{Average black pixels} = \frac{1}{N} \sum_{p \in \mathcal{P}} \Pi$$

Where  $\mathcal{P}$  is the pixels in the image,  $N$  is the number of images, which is 150 thousand images in our work and  $\Pi$  as the following:

$$\Pi = \begin{cases} 1 & \text{if } p = 0 \\ 0 & \text{else} \end{cases}$$

## Appendix B

In our work, we employed an NVIDIA RTX 4090 graphics card, an Intel Core i9-14900K CPU, and 64 GB of RAM operating at 4000 MT/s.