THE DEPTH PERCEPTERS
CIS 520 Final Project Report

Charles Baird Howland [1] **Athrva Pandhare** [2] **Sowmya Sridharan** [3] **Frederick Xu** [4]

Abstract

Depth perception is the visual ability to perceive the world in 3D, and is an essential step for tackling vision problems in fields such as robotics, autonomous driving, and tasks involving 3D reconstruction. In the case of monocular visual systems–increasingly in the case of real-world applications–depth perception is difficult because stereoscopic approaches are unavailable, and depth has to be inferred from a single image. Here we test two supervised approaches for learning the corresponding depth maps to an input RGB image: a baseline UNET with Mean Square Error loss, and a novel GAN model which uses the same UNET architecture as the generator and is ultimately also evaluated with MSE loss. After training on around 4000 examples and evaluating on a validation set, we found that the GAN model, with a loss of 0.0005, outperforms the baseline UNET model whose loss is 0.035.

1. Introduction and Motivation

Intuitively, depth perception is the sense of how close we are to the various objects in our field of view. Since the optical images created by visual systems–whether the animal's eyes or the camera–are inherently two-dimensional, depth perception is a matter of inference from two-dimensional images to a three-dimensional perception of the world.

Traditional depth perception techniques assume a binocular visual system (i.e., two lenses) and rely on what are called stereoscopic cues, the comparison of two images of the same scene from slightly different angles. In recent years, the field has concentrated on the more difficult task of monocular depth perception, where depth cues are limited to whatever information is in a single image, such as relative size of objects, lighting, texture, elevation, and any other visual patterns. The focus on monocular depth perception has come about in part because it is increasingly the case in real world applications that only one camera is available to capture images.

Here, we we compare the performance of two monocular depth perception systems trained and validated on the KITTI Vision Benchmark dataset. Our baseline approach is the standard UNET architecture trained with MSE loss, while the more novel approach makes use of a GAN, where the generator is the same UNET from the baseline and the discriminator is a Patch-GAN trained from scratch.

2. Related Work

A recently published review article in 2021 described previous approaches to monocular depth perception (Ming et al., 2021). Previous work has made use of a range of contemporary deep learning techniques, including deep image networks such as VGG-16, AlexNet, and ResNet in supervised learning regression tasks, where skip connections and up-convolutions, akin to a UNET structure are used to create depth maps.

Depth Perception can be viewed as an image generation task, which is often accomplished using an encoder-decoder architecture that outputs generated images. Previous studies noted that training encoder-decoder networks with L1 or L2 penalties often leads to blurry images (Pathak et al., 2016) (Zhao et al., 2017). This is believed to happen because such naive loss functions do not penalize the networks on the joint configuration of the output pixels. A promising alternative approach makes use of GANs (Pathak et al., 2016). GANs are a special type of neural network architecture composed of two deep nets, called a generator and a discriminator. The generator is generally an encoder-decoder architecture that produces an image (given either a latent vector as an input or another image as an input). The discriminator discriminates between the synthetic images generated by the generator and the ground truth images. Ideally, in the training process, the two networks improve together (i.e. the generator gets better at producing images similar to the ground truth, and the

[1] baird.howland@asc.upenn.edu (bairdh) [2] athrva@seas.upenn.edu (athrva) [3] sowmya@seas.upenn.edu (sowmya) [4] fredxu@seas.upenn.edu (fredxu).

discriminator gets better at discriminating between synthetic images and the true images). GANs, therefore, propose an adaptive loss function that can impose coherence conditions on the output images. It is also believed that combining a high-level adaptive loss of a GAN with a low-level L1 loss may boost performance even further. We propose the use of GANs for the monocular depth perception task, since it requires that the global contextual information be extracted from the input view.

3. Dataset

KITTI (Karlsruhe Institute of Technology and Toyota Technological Institute) is one of the most popular datasets for applications in mobile robotics and autonomous driving. It consists of hours of traffic scenarios recorded with a variety of sensor modalities, including high-resolution (1392 x 512) RGB, grayscale stereo cameras, LiDAR, and a 3D laser scanner. We use the annotated depth maps dataset alongside available monocular RGB raw data, which comes in the form of image sequences (around 100 images per set) taken from a camera mounted to a car. The depth maps–the ground truth in our case–number around 100,000 and are 14GB, but due to practical resource and time constraints, we trained on only 3824 image-depth map pairs. Since we are using deep learning approaches, our feature set is not defined explicitly. We instead learn the features using convolutional network layers.

4. Problem Formulation

It is worth stating again that the depth perception performed by animals' visual systems is not fundamentally the result of learning to see how depth "looks" in a given image. Rather, animals have two eyes, and the depth of various objects in a visual field can be inferred from the subtle differences in the images received by the two eyes, whose separating distance is known. When–as is the case with many real-world examples–a visual system only creates images from a single vantage point, traditional strategies in animals' visual systems fail as a guiding metaphor and an alternative approach of learning depth directly from a single image is necessary.

Here, we will explore a supervised machine learning approach, where the system learns how depth "looks" by considering many examples of images whose ground truth depth map is known. Specifically, the model input is a 2D RGB image, and the output is a depth (distance from the camera to the surface) prediction for every pixel in the input image: a depth map. Convolutional deep learning approaches will learn various representations of the input image which reflect the depths in the ground truth. One aspect of the machine learning task which is somewhat atypical is that the model output is not a single regression or a distribution over categories but an estimate of depth for every pixel in the input image–in essence another channel.

5. Methods

5.1. Data Pre-processing

The data made available in the KITTI dataset contains RGB images of the scene, the corresponding projected raw LIDAR scans and the odometry data. In this project, the task can be thought of as image translation from a form $\mathcal{A}$, the RGB image, to a form $\mathcal{B}$, the depth-map. As stated previously, the data provided in the KITTI dataset includes the projected point clouds which have to be converted into the respective depth-maps manually. This is also known as depth filling and is a complete research field in it's own right. For our purposes, we use the method suggested in (Ku et al., 2018) to fill the sparse projected LIDAR scans. The Depth inpainting process is guided by the RGB images. It could be thought of as laying the projected LIDAR scans on the RGB image and using the dense structure of the RGB image for depth filling.

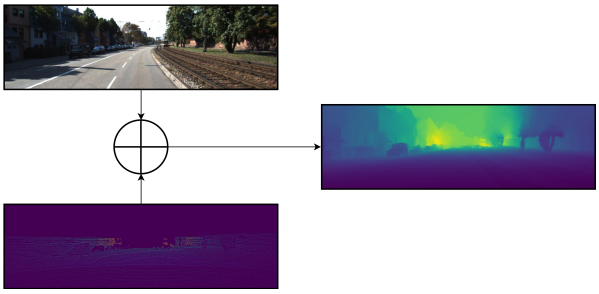

Figure 1. Depth filling pipeline

Images were also downscaled to 416x128 due to VRAM concerns on Google Colaboratory.

5.2. Baseline Model: Pretrained UNET

As depth maps require pixel-level predictions, it follows that many approaches follow an encoder-decoder like structure. One such approach is the UNet, which has been used for semantic segmentation (Ronneberger, 2017). We chose this UNet structure for our baseline, as an implementation is readily available in PyTorch with open-source code (Github repository). Since we require our values to be clamped between 0 and 1 for skimage RGB format, we elected to keep the sigmoid activation function in the UNet model.

To change from a classification task to a regression task, we require a change of loss function. Commonly used loss functions for supervised depth perception are log depth and

Berhu which combines an L1 and L2 loss (Table 1) [1]. In these equations, d represents the estimated depth and d* represents the ground truth depth. However, we elected to use the mean square error loss as a baseline due to ease of availability in PyTorch.

Table 1. Loss functions

Name	Formulation
$L(\log d)$	$L\left(d, d^*\right) = \frac{1}{N}\sum_{i=1}^{N} y_i^2 - \frac{\lambda}{N}\left(\sum_{i=1}^{N} y_i\right)^2$
$Berhu$	$L_{Berhu}\left(d, d^*\right) = \begin{cases} \lvert d-d^*\rvert & \text{if } \lvert d-d^*\rvert \leqslant c \\ \frac{\lvert d-d^*\rvert^2+c^2}{2c} & \text{if } \lvert d-d^*\rvert > c \end{cases}$

We trained the model for 10 epochs on Google Colaboratory, using 60% of the data in the training set, 20% for validation, and 20% for testing, with batch sizes of 16. We used Adam optimizer with a learning rate of 1e-5, a step size of 10, and γ of 0.1, and report the loss at each batch as well as at the end of training.

5.3. GAN Loss Function

In this section we define the composite loss function used for training the GAN. The loss function consisted of the following components.

1. The *Gradient loss*, which is a novel implementation. Adding this component of the loss ensures that the edges are conserved in the generated images. We found that it sped up the training process significantly.

2. *Cross-entropy loss* between the discriminator predicted patches and the adversarial ground truths. This component of the total loss allows the generator to learn based on the feedback provided by the discriminator.

3. L_1 *Loss* between the discriminator predicted patches and the adversarial ground truths. This component of the total loss also allows the generator to receive feedback signal from the discriminator, additionally, it allows for a more fine grained control of the discriminator induced changes in the generator's training.

4. L_1 *Loss* and *MSE Loss* between the generated depth map and the ground truth depth map. This component of the total loss is a direct loss, i.e., it has a direct bearing on the generated depth map. Adding this loss speeds up the initial learning process when both the generator and the discriminator are untrained. We now present a mathematical formulation of the loss functions used in the GAN training process.

5.3.1. The Gradient Loss

Consider that $\mathcal{R}$ is the input RGB image of dimensions $(M \times N)$ and $\mathcal{I}$ is the corresponding ground truth depth map of the same dimensions.
In general, we define a convolution operation using a kernel $\mathcal{K}$ (of dimensions $(m \times n)$ as,

$$\mathcal{O}_{i,j} = \sum_{k=1}^{M}\sum_{l=1}^{M} \mathcal{I}(i+k-1, j+l-1) \times \mathcal{K}(k,l) = \mathcal{I} \circledast \mathcal{K} \quad (1)$$

where i runs from 1 to $M - m + 1$ and j runs from 1 to $N - n + 1$. Let g_x and g_y be the horizontal and vertical gradient kernels respectively. Then the horizontal and vertical gradient of the ground truth depth map is given by,

$$\begin{aligned} G_x &= \mathcal{I} \circledast g_x \\ G_y &= \mathcal{I} \circledast g_y \end{aligned} \quad (2)$$

Similarly, the x and y gradient of the generated depth map $\hat{\mathcal{I}}$ is given by,

$$\begin{aligned} F_x &= \hat{\mathcal{I}} \circledast g_x \\ F_y &= \hat{\mathcal{I}} \circledast g_y \end{aligned} \quad (3)$$

Now, we define the bounded gradient image of the ground truth depth-map $\mathcal{I}$, and the generated depth-map $\hat{\mathcal{I}}$ as G_{xy} and F_{xy} respectively.

$$\begin{aligned} G_{x,y} &= S\left\{\frac{G_x^2}{\max(G_x^2)} + \frac{G_y^2}{\max(G_y^2)}\right\} \\ F_{x,y} &= S\left\{\frac{F_x^2}{\max(F_x^2)} + \frac{F_y^2}{\max(F_y^2)}\right\} \end{aligned} \quad (4)$$

where S is the sigmoid function.

Finally, the gradient loss may be formulated as follows,

$$\nabla_{loss} = \sum_{i=1}^{M-m+1}\sum_{j=1}^{N-n+1} ||G_{x,y}| - |F_{x,y}|| \quad (5)$$

5.3.2. Formulating the Generator Loss

Let us define the generator loss as $\mathcal{G}(\hat{\mathcal{I}}, \mathcal{I}, \mathcal{R}, G_{x,y}, F_{x,y}, \mathbf{D}, \mathbf{1}_{adv})$ which we define as $\mathcal{G}$ for concise notation. We define $a_1, a_2, a_3, a_4 \in \mathbb{R}$ (representing the weights assigned to each of the components in the total loss), L_1 denoting the L_1 loss and L_2 denoting the L_2 loss. Finally, $\mathbf{1}_{adv}$ and $\mathbf{0}_{adv}$ are the adversarial ground truths. Then we can define the total Generator loss as,

$$\begin{aligned} \mathcal{G} &= a_1 \times \nabla_{loss} + a_2 \times L_1(\hat{\mathcal{I}}, \mathcal{I}) + a_3 \times L_2(\hat{\mathcal{I}}, \mathcal{I}) \\ &+ a_4 \times \mathcal{C}(\mathbf{D}(\hat{\mathcal{I}}, \mathcal{R}), \mathbf{1}_{adv}) \end{aligned} \quad (6)$$

where $\mathcal{C}$ is the Cross-entropy loss.

5.3.3. Formulating the Discriminator Loss

The discriminator loss is given by the following equation,

$$\mathcal{D} = L_2(\mathbf{D}(\mathcal{I}, \mathcal{R}), \mathbf{1}_{adv}) + L_2(\mathbf{D}(\hat{\mathcal{I}}, \mathcal{R}), \mathbf{0}_{adv}) \quad (7)$$

This loss function has the effect of teaching the discriminator that the real ground truth depth map corresponds to the $\mathbf{1}_{adv}$ tensor, and the fake/generated depth map corresponds to the $\mathbf{0}_{adv}$ tensor.

The GAN's training process then, can be described by a combined minimization problem depicted in equation (8),

$$\min \mathbf{w} = (\min_{\mathcal{G}} \mathbf{G}, \min_{\mathcal{D}} \mathbf{D}) \quad (8)$$

6. Experiments and Results

6.1. UNet

The UNet training was overall successful, with the loss decreasing steadily (Figure 2a). However, much of the gain occurred in the first epoch, before the loss decreased linearly. The linear decrease in loss of the validation can be also observed, as the validation loss was only recorded from the first epoch onward (Figure 2b). For final losses, the validation set performed the best at 0.0346, followed by training set at 0.0369, and testing set at 0.0408 (Table 2). As such, we can observe that the model has not overfit on the training set, in fact it performs the best on the validation. Losses are comparable between the three sets.

Examples of UNet outputs of the the training and validation set were sampled at the end of the 1st, 5th, and 10th epochs of training (Figure 5). We can see that the UNet depth outputs are very noisy, and appear to lack detail overall. We cannot really discern individual objects very well, and can only discern very large elements in the image like the road and the outline of the trees. However, it can be seen that the UNet is capable of learning depth, with the background being properly identified, and increase in green intensity with epochs to indicate that the UNet is perceiving greater and greater depth with each epoch (Figure 10). Although the results have some promise, it appears that this baseline method is unable to recapture the original depth map accurately.

Table 2. Loss values for methods

Method	Train	Validation	Test
UNET, MSE	0.0369	0.0346	0.0408
GAN	0.0061	0.0005	0.0047

6.2. GAN Training and Results

The GAN training process requires a balance between the training of the Generator and the Discriminator. If either model overpowers the other by a significant amount, the learning can essentially come to a halt. We therefore employed a progressive learning process which included changing the relative learning rates and relative update frequencies of the Generator and Discriminator during training. We illustrate this using the MAE loss curve in Figure 3. At point **A**, the learning rate of the Discriminator was halved and the update frequency was changed to once in five batches. The reason for making this change was that the discriminator was outperforming the generator significantly, and causing unstable behavior in the training framework. If left unchecked, this could lead to divergence; where the discriminator severely outperforms the generator causing the learning process to halt. As is evident from Figure 3, this had a positive effect on the training process (there was a notable drop in the loss). The point **B** denotes an inflection point where the loss starts to plateau. After this point any further drop in the loss would require a corresponding decrease in the discriminator loss. This implies that both the agents (i.e., the generator and discriminator) need to improve in adversarial training for the overall loss to decrease further. From an optimization perspective, this could signify an escape from a local optimum. From here onward, in all GAN loss curves, we show points **A** and **B** as a reference.

Figures 4 and 5 show the MSE as well as total generator loss. Recall that the total generator loss is a composite quantity which includes the gradient loss defined in section

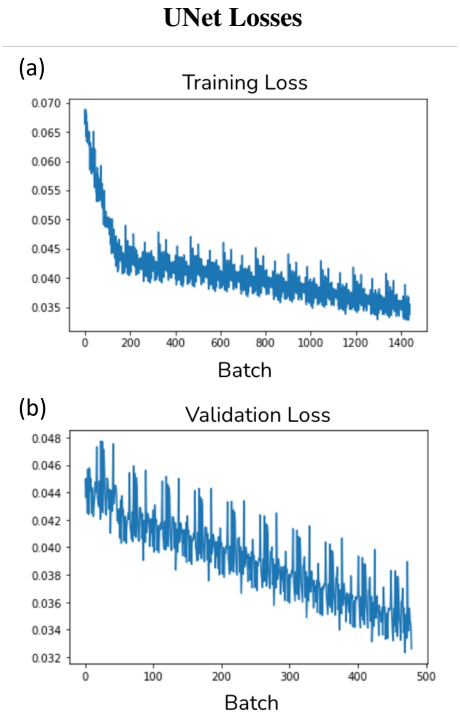

Figure 2. UNet Training and Validation Losses per batch

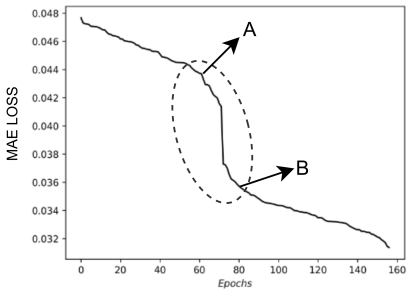

Figure 3. Generator MAE loss for the training dataset

5.3.1. We observe similar trends in Figures 4 and 5, with the exception that the decrease in the losses at point **A** are not as drastic. We now discuss the validation loss curves for

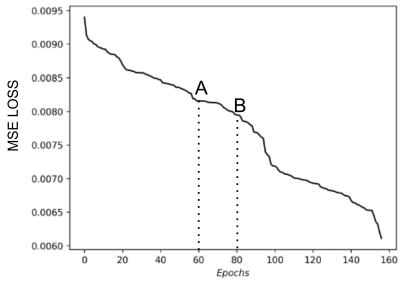

Figure 4. Generator MSE loss for the training dataset

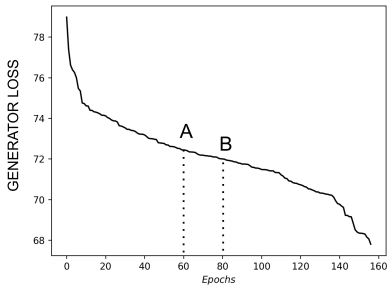

Figure 5. Total Generator loss for the training dataset

the GAN. The validation loss was tracked throughout the training process, and the effects of reducing the discriminator's learning rate and update frequency are apparent from the loss curves. Figure 6 shows the MAE loss curve for the validation data. A similar and perhaps more drastic decrease is seen at point **A** which reaches a final inflection point at **B**, after which the loss reaches a plateau. Figures 7 and 8 show the generator's MSE loss and total loss on the validation dataset. We can observe a similar trend in Figures 7 and 8, where the training rate is accelerated by tweaking the dis-

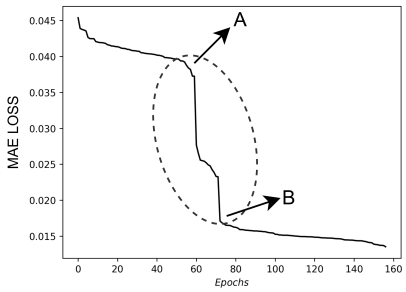

Figure 6. Generator MAE loss for the Validation dataset

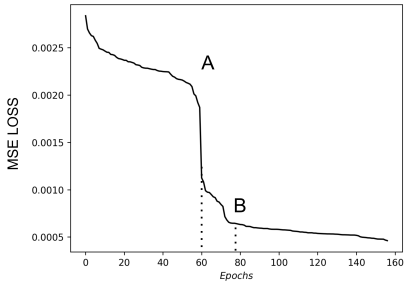

Figure 7. Generator MSE loss for the Validation dataset

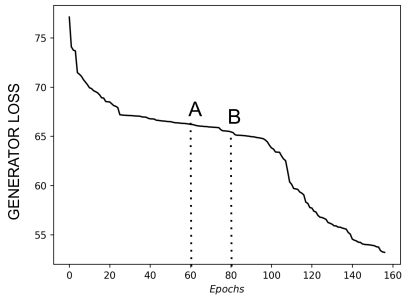

Figure 8. Generator total loss for the Validation dataset

criminator's learning rate and update frequency. We further notice that the GAN reaches a much lower overall MSE loss as compared the the UNet model. Note that the architecture used here in the generator is also the UNet architecture. This choice was made to make comparison easier. This further implies that the difference in results is a direct consequence of the type of training and the loss functions employed (adversarial with gradient loss for GAN and static MSE for UNet baseline).

7. Discussion

Firstly, our baseline model, the UNet, performed less than optimally. It can be observed that the depth maps produced

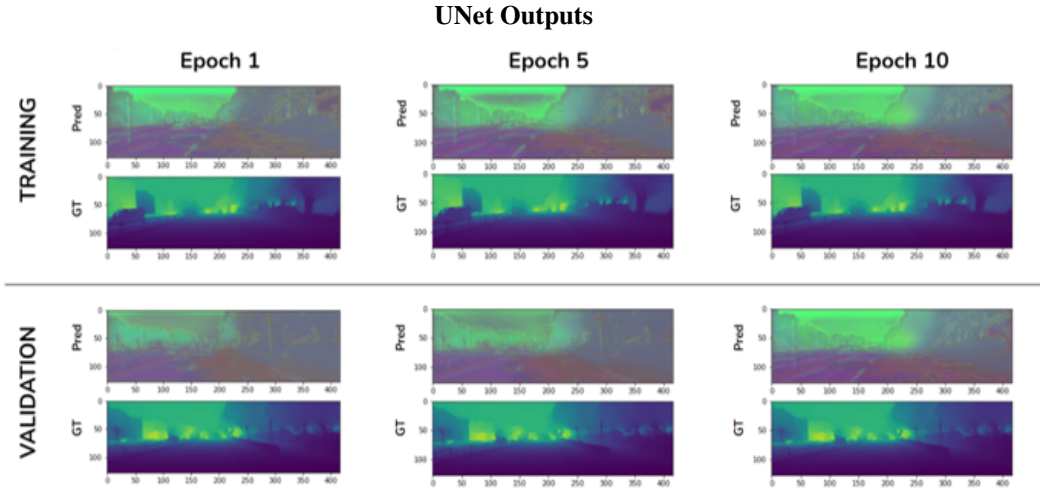

Figure 9. Output for the UNet on the 1st epoch

by the model output are quite noisy, and the objects in the environment are not captured properly in the depth map. This is especially problematic, as one primary use of depth maps in modern technology is for object detection based on proximity. Previous works have made modifications to the UNet structure, introducing additional convolutional layers in the skip connections (He et al., 2018) or training on an additional gradient stream (Li et al., 2017). Additional considerations include the choice of loss function. As mentioned previously, there exists mathematical challenges where the loss can be unstable, thus leading to formulations based on combined losses (Ming et al., 2021). Lastly, it may also be the case that the UNet structure itself is not well-suited for the task, as its original design was intended for semantic segmentation, and thus its structure was not tuned for fine-grained pixel-level predictions. The task may be better suited for a dedicated generative model. As such, to address these innovations, we chose to experiment with a GAN while combining a variety of loss functions including a loss function based on the gradient of the image.

The GAN performed better than our baseline model in all the recorded metrics. We attribute this primarily, to the adversarial training and the composite loss function. Figure (10) shows the images generated using the GAN. It is clear that the depth is correctly estimated in almost all cases. Owing to the addition of the gradient loss function, the generator produces distinct edges between objects in the depth-maps. The adversarial training allows the generator to learn not only an average representation of the depth-maps, but the underlying distribution itself, from which the depth maps have been sampled (at least in the domain of the data samples).

From an implementation perspective, we were able to deduce important insights during the GAN training process. Primarily the balance on the relative performances of the agents in adversarial training (generator and discriminator). We further investigated the effects of asymmetric learning rates assigned to the generator and the discriminator. We found that such an approach was beneficial for our case. However, we think that the specifics of the training process are a heuristic for our specific problem, and while the principle of unequal learning rates and update frequencies is helpful, the specifics of the training process are only applicable to our training process. Furthermore, we found that the training process depends heavily on the choice of the discriminator and generator models.

7.0.1. A Note on the Gradient Loss

The gradient loss proved to be extremely beneficial to the training process. To that end, we would like to elaborate further on it's effects on the produced images. The gradient loss function penalizes difference in the gradients between the generated depth-map and the ground truth depth map. An obvious effect of this is that it encourages the generator to learn the edges/boundaries of the depth maps correctly. In a depth-map, perhaps the second most important property to predict after the pixel intensities are the edges, which imply the ability of the generator to recognize multiple objects.

As a note on the implementation side of the gradient loss, it is very important to adjust the relative weight (a_i) assigned to the gradient loss. This needs to be done to ensure that the gradient loss does not overpower the other components of the total loss function.

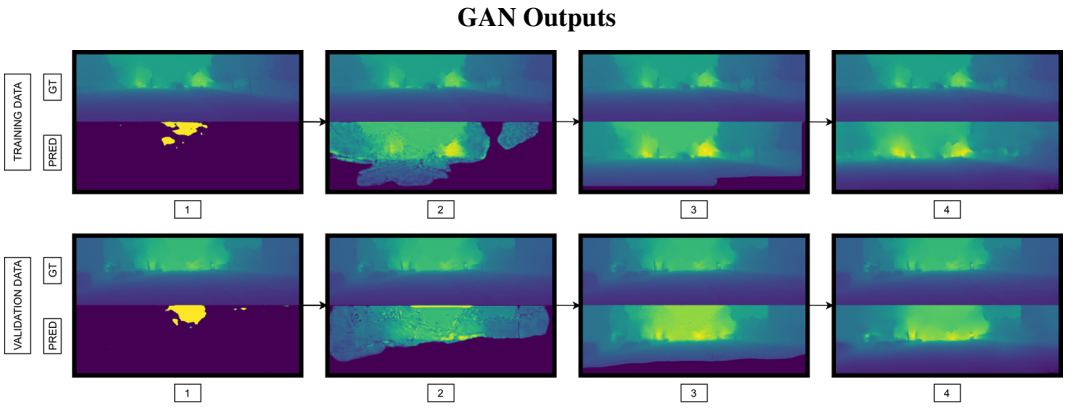

Figure 10. GAN Outputs spanning the entire training process

8. Conclusion

For the task of monocular depth perception, adversarial training improves performance of a generative Convolutional Neural Network. For this project, we used a relatively small dataset of 3500 image pairs. We were nonetheless able to prove the superiority of adversarial training for monocular depth estimation. With that said, there are further improvements possible. Both in the training and design of the baseline model and the GAN. In the future, we will investigate the models using a larger set of training data, as well as test the models on the NYU Vision dataset which is another popular benchmarking dataset. For the UNet baseline, we will revisit with implementations of the BerHu loss, as well as replicate experimental models presented in previous work. For the GAN model, we intend to shift from a pix2pix type implementation to a cycleGAN type implementation. The advantage of this is that the training data does not need to be pairwise (by treating depth and RGB to be two representations of images).

9. Acknowledgement

The authors especially thank the University of Pennsylvania CIS 520 Machine Learning instructors, Prof. Lyle Ungar and the TAs, for their advice and support throughout this project and the semester.

References

He, L., Wang, G., and Hu, Z. Learning depth from single images with deep neural network embedding focal length. *IEEE Transactions on Image Processing*, 27(9): 4676–4689, 2018. doi: 10.1109/tip.2018.2832296.

Ku, J., Harakeh, A., and Waslander, S. L. In defense of classical image processing: Fast depth completion on the cpu. *2018 15th Conference on Computer and Robot Vision (CRV)*, 2018. doi: 10.1109/crv.2018.00013.

Langley, P. Crafting papers on machine learning. In Langley, P. (ed.), *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pp. 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.

Li, J., Klein, R., and Yao, A. A two-streamed network for estimating fine-scaled depth maps from single rgb images. *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017. doi: 10.1109/iccv.2017.365.

Ming, Y., Meng, X., Fan, C., and Yu, H. Deep learning for monocular depth estimation: A review. *Neurocomputing*, 438:14–33, 2021. doi: 10.1016/j.neucom.2020.12.089.

Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., and Efros, A. A. Context encoders: Feature learning by inpainting. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. doi: 10.1109/cvpr.2016.278.

Ronneberger, O. Invited talk: U-net convolutional networks for biomedical image segmentation. *Informatik aktuell*, pp. 3–3, 2017. doi: 10.1007/978-3-662-54345-0_3.

Zhao, H., Gallo, O., Frosio, I., and Kautz, J. Loss functions for image restoration with neural networks. *IEEE Transactions on Computational Imaging*, 3(1):47–57, 2017. doi: 10.1109/tci.2016.2644865.