Depth Sense : A monocular approach to estimate relative Depth

A Final Project for DLIV 2023

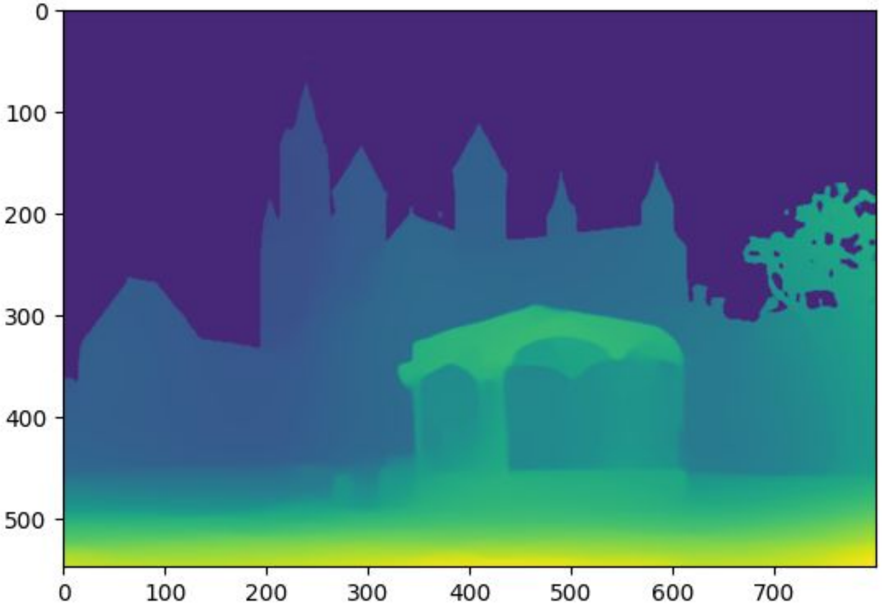
-Aditya Shourya (i6353515)

Maastricht University

Quick Recap

Problem statement : To introduce a simple , effective and an unbiased method to automatically generate dense relative depth annotations from a given monocular image.

Depth Mapping



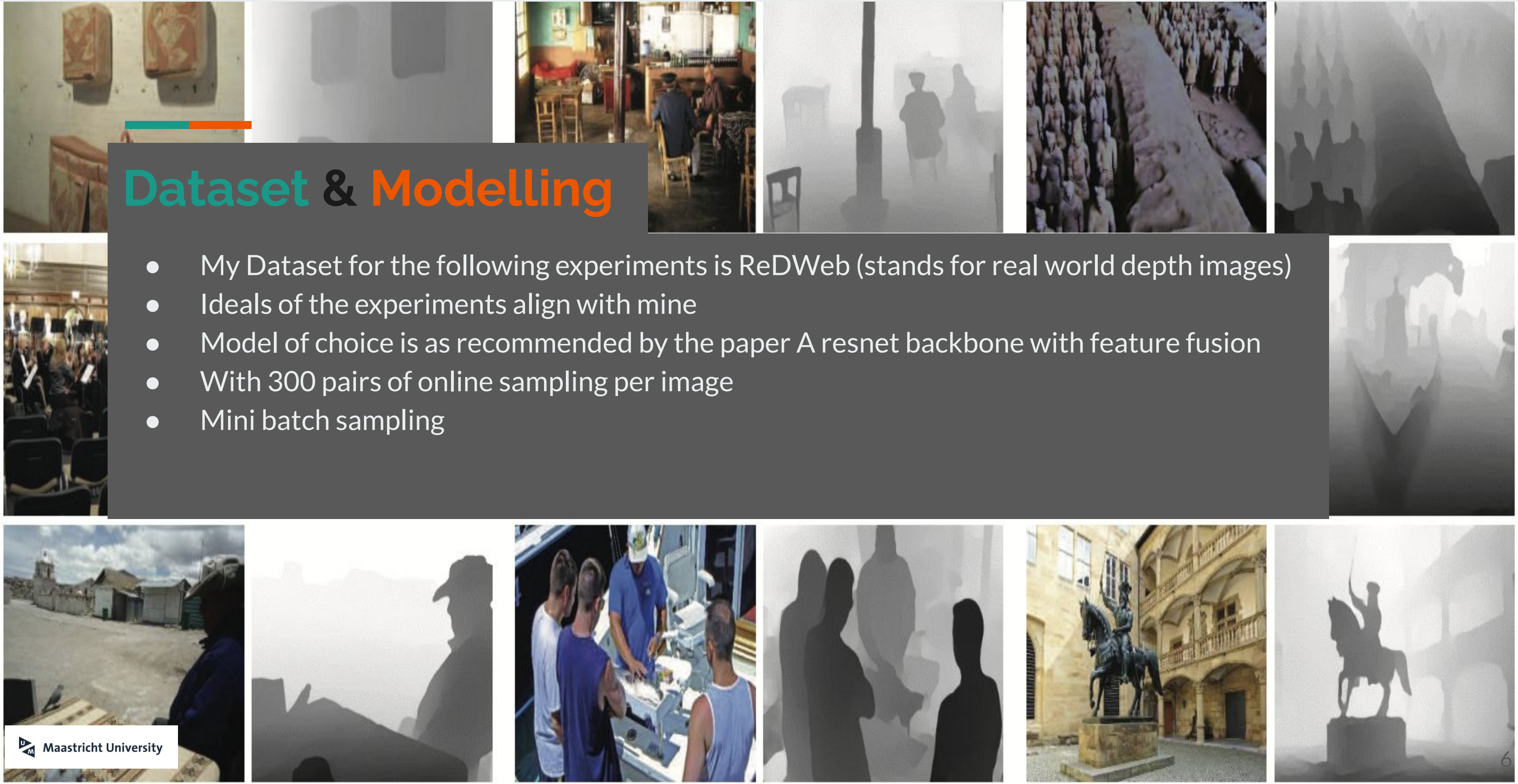
Left : A Monocular image of Vrijthoff (which insinuates of a 3d setting) ; Right : pixel intensity difference implies the relative depth. Produced by intel's MiDas (Current state of the art model to get relative depth from monocular images or even a live video feed)

Code Navigation

- Where to find my code
- Code Report
- Academic Report
- Notebooks , scripts
- Model Weights

Project Significance

- The objective of the project is not to improve on accuracy or computational efficiency of any existing established models.
- But the aim is to experiment and understand the components and working of different geometrical losses
- And to device a combination of losses into one such that it is unassuming and does not pollute the graph of the network by bringing redundancy
- Practically should be applicable to all kinds of images from the web



Dataset & Modelling

- My Dataset for the following experiments is ReDWeb (stands for real world depth images)
- Ideals of the experiments align with mine
- Model of choice is as recommended by the paper A resnet backbone with feature fusion
- With 300 pairs of online sampling per image
- Mini batch sampling

Maastricht University

Mini Batch Sampling

Figure 4: Consider this example of a horse in a museum along side its depth map . to calculate ordinal relations between different manifolds of the image we choose many areas at random and refer to its ground truth to caculate ordinal relations

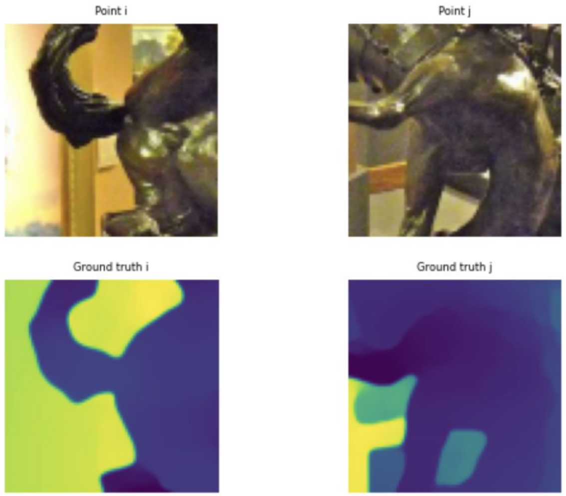
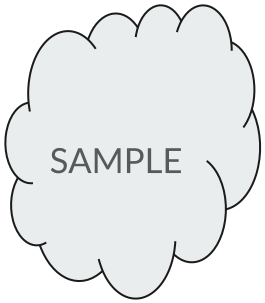


Figure 5: Here we focus on one such sampling pair to get our ordinal relation. here the ordinal relation is the difference between the average pixel value with it's corresponding ground truth.This essentially converts our problem into solving a regression equation



- Mini-batch transforms the inherent complexities of depth estimation into a more manageable regression problem.
- This process leverages the concept of sampling ordinal relations to guide the model towards understanding depth variations.

Feature Fusion with Resnet Backbone

A Brief understanding of what Resnet Backbone with feature fusion would work in its forward pass :

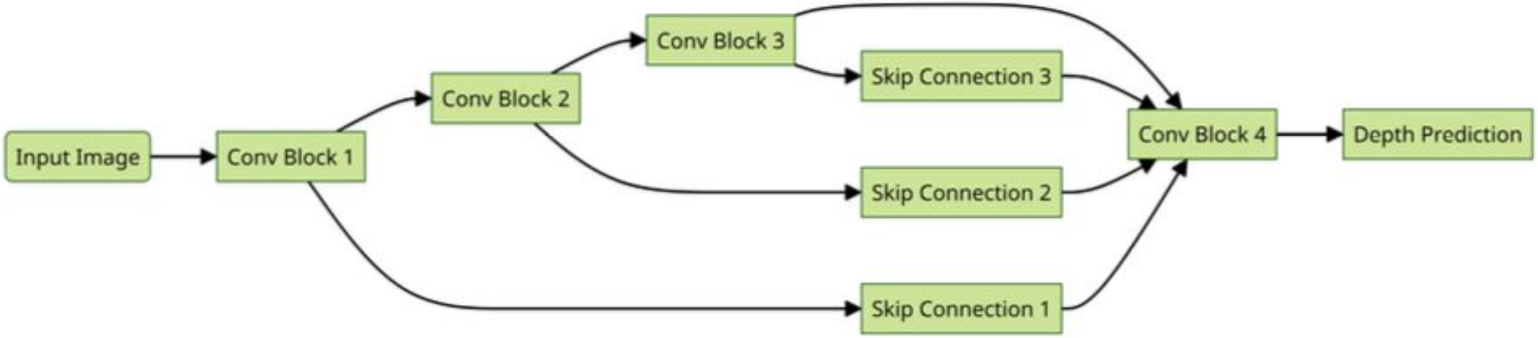


Figure 6: A Block Diagram representing how skip connections (feature fusion) works in Neural Networks

Geometrical Losses

- Structural Similarity Index Loss
- Inverse Depth Loss Smoothness
- (Self-Made)Edge Guided loss with combination

Structural Similarity Index

$$SSI(P,G) = \frac{2\mu_P\mu_G + c_1}{\mu_P^2 + \mu_G^2 + c_1} \cdot \frac{2\sigma_{PG} + c_2}{\sigma_P^2 + \sigma_G^2 + c_2} \tag{1}$$

Here, the terms are defined as follows:

- μ_P and μ_G are the means of P and G, respectively.
- σ_P and σ_G are the standard deviations of P and G, respectively.
- σ_{PG} is the covariance of P and G.
- c_1 and c_2 are constants to stabilize the division with a weak denominator.

Figure 2: This figure shows how this example is less than ideal for SSI Loss . Although there is a lot of structure that needs preserving the luminance remains almost the same as its background .A good example pair would be to use them when the prominent subject has higher luminescence and is well separated with contrasting edges.

- **Luminance** (l)**:** Represents the average pixel intensity. It measures the overall brightness of the images.
- **Contrast** (c)**:** Captures the standard deviation of pixel intensities, reflecting the variation in brightness. High contrast indicates a wide range of intensities.
- **Structure** (s)**:** Describes the covariance of pixel intensities between P and G. It assesses how patterns in pixel intensities are related between the two images.

Updated to work with current pytorch #45

Edit | Code

Open adishourya wants to merge 1 commit into `Po-Hsun-Su:master` from `adishourya:master`

Conversation 0 | Commits 1 | Checks 0 | Files changed 1 | +19 −33

adishourya commented now

I have used and tested this with my redweb dataset (rgb iamge and depth map pair) seemed to work fine.

I am open to do more tests on this if you need me to

Best,
Aditya

`Updated to work with current pytorch` ... Verified c83f7bf

Add more commits by pushing to the `master` branch on **adishourya/pytorch-ssim**.

This branch has no conflicts with the base branch
Only those with write access to this repository can merge pull requests.

Add a comment

Write | Preview

Add your comment here...

Reviewers
No reviews
Still in progress? Convert to draft

Assignees
No one assigned

Labels
None yet

Projects
None yet

Milestone
No milestone

Development
Successfully merging this pull request may close these issues.

None yet

Inverse Depth Smoothness Loss & Custom Edge guided Loss

- You can find more about Inverse Depth Smoothness loss in the academic paper. But I use Techniques from Inverse Depth Smoothness loss in my custom Edge guided loss to finally make a loss function .
- My loss function aims to balance the effects of equal and unequal depth ratios using a hyper parameter.

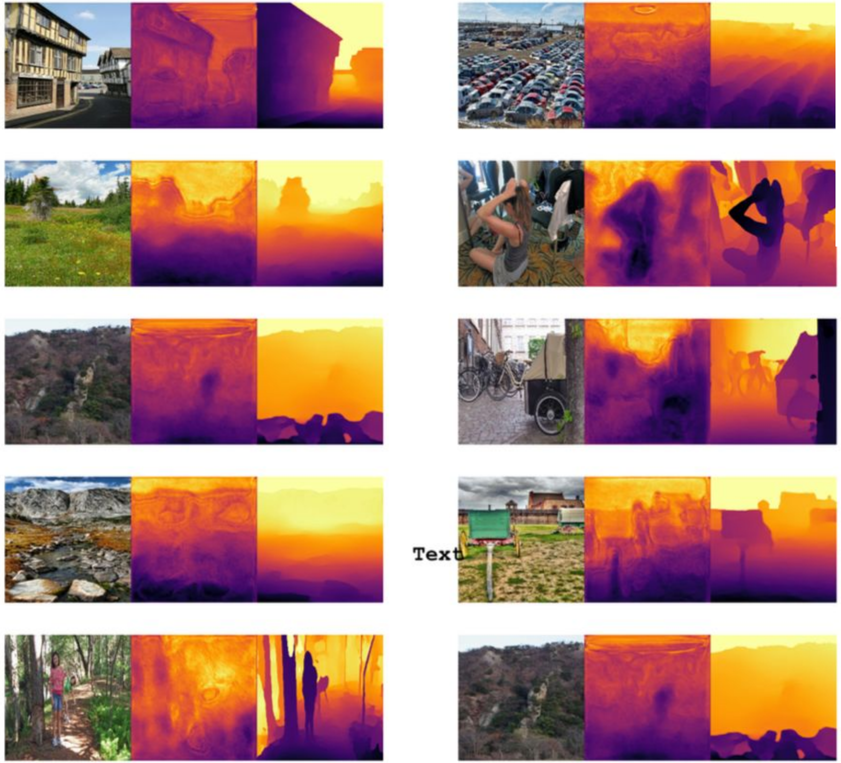
Improving on mini batch sampling:

- Positive (significant) pair selection from mini batch sampling
- Consistent mask selection

Masks and edge guidance. Features introduced to masks:

- Edge detection
- Edge point , anchor point selection
- Generate 4-point coordinates
- Brings Regularization

Results & Validation



	loss_train	loss_val	ssim_train	ssim_val	mse_train	mse_val
0	0.327318	0.2071	0.133579	0.425158	0.327318	0.20583
1	0.050247	0.042086	0.58262	0.608989	0.050247	0.041778
2	0.054153	0.047912	0.647623	0.622574	0.054153	0.047796
3	0.047921	0.039307	0.680513	0.654263	0.047921	0.039069
4	0.043474	0.038338	0.688893	0.655723	0.043474	0.038063

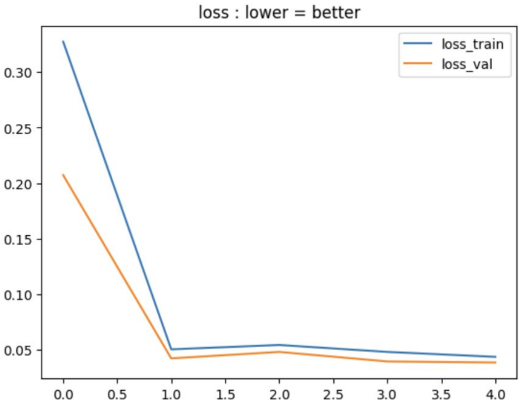


Figure 8: Best Epoch output. $\text{loss}_f n = EdgeGyudideLoss()$

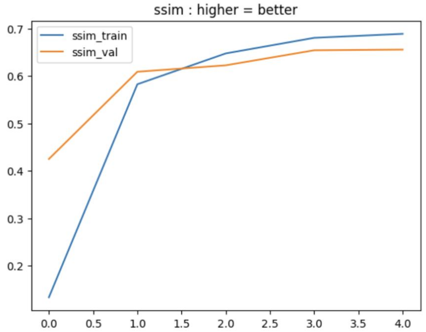


Figure 10: Structural Similarity Index performance

Understanding our Model’s limitation & Its ideal scenarios

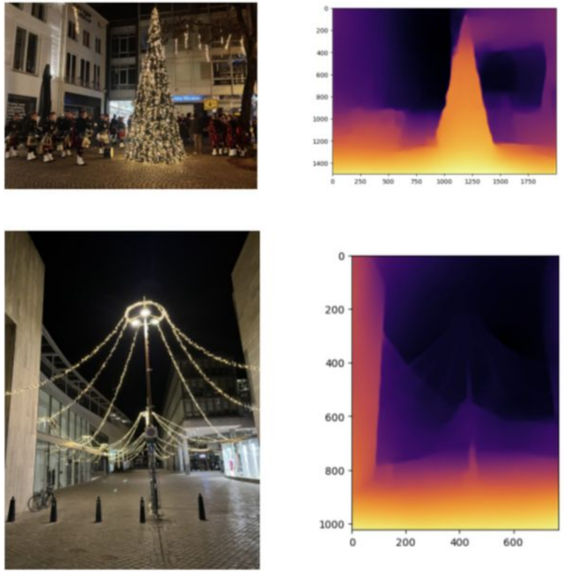


Figure 12: **Top image**: you can see that model does enough to detect the christmas tree and maintains a balance in depth smoothness when we look closely at the band surrounding the tree . which hints at the claim that judicially combining geometrical losses can improve on the quality and usability of depth perception . **Bottom Image** : since the main subject is too thin the chances of genrating ordinal distances from the lampost decreases quite a lot and almost fails to capture its complete structure

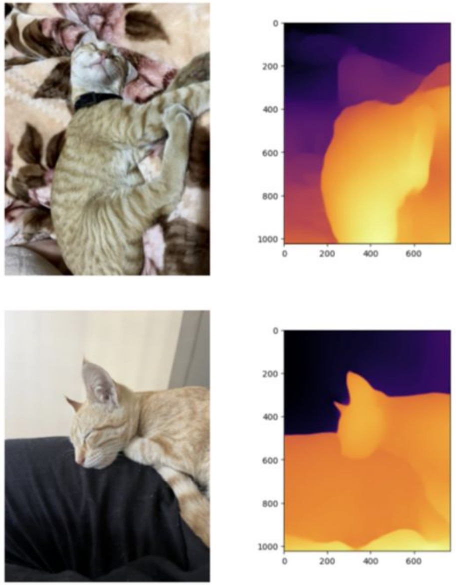


Figure 13: **Top Image** : Since the subject is close enough to the camera and does not inherit a lot of structure the model performs well on preserving smoothness. **Bottom Image** : A Cherry-picked image of my cat shows the ideal scenario when we can get by using a smaller parameter model/ high inference for tasks that demands less of the structure

Thankyou

Acknowledgement

Research Contributions:

- Xian et al., 2018: "Monocular Relative Depth Perception with Web Stereo Data Supervision" presented at CVPR 2018.
- Saxena et al., 2009: "Learning Depth from Single Monocular Images" from NIPS 2009.
- Hirschmuller et al., 2005: "Real-Time Stereo Matching Using Adaptive Window-based Dynamic Programming" from CVPR.
- Zbontar and LeCun, 2016: "Stereo Matching by Training a Convolutional Neural Network to Compare Image Patches" in the Journal of Machine Learning Research.
- Vogel et al., 2015: "Displets: Resolving Stereo Ambiguities using Object Knowledge" presented at CVPR 2015.
- Pang et al., 2017: "Efficient Deep Learning for Stereo Matching" from CVPR 2017.
- Laina et al., 2016: "Depth Prediction Without the Sensors: Leveraging Structure for Unsupervised Learning from Monocular Videos" at CVPR 2016.
- Chen et al., 2016: "Single-image depth perception in the wild" from NeurIPS 2016.

Open Source Contribution:

- SSI Loss https://github.com/Po-Hsun-Su/pytorch-ssim/pull/45

Inspiration for Deep Learning Architectures:

- He et al., 2016: Acknowledgment to "Deep Residual Learning for Image Recognition" for providing inspiration for deep learning architectures.

References:

- The references section includes the citations and links to the respective papers, repositories, and resources mentioned above.

Special Thanks:

- Express gratitude to mentors, colleagues, and anyone else who contributed to the success of the project.

Project Contributors:

- Acknowledge the members of your project team and their individual contributions.