#*****

Monocular Depth Estimation

Raj Anandak, Rajnish Gupta

Abstract

Monocular depth estimation from single RGB images has made substantial progress with advanced techniques involving Vision Transformers, Attention Models, and Recurrent Neural Networks, which provide robust depth estimation with temporal consistency. However, our project operates within the constraints of limited computational resources and project scope. Consequently, we opt for a simpler encoder-decoder architecture to strike a balance between performance and feasibility. Our focus is on enhancing the quality and boundary precision of depth maps using conventional neural network architectures. We leverage transfer learning, incorporating image encoders originally designed for image classification, which retain spatial resolution and employ skip connections to enhance depth estimation. Additionally, we explore the potential of incorporating camera intrinsic parameters for improved depth estimation. The DIODE dataset serves as our resource, though it lacks consecutive frames. To address this limitation and enhance temporal consistency, we plan to retrain our model on custom datasets generated from depth-sensing cameras, thereby making it applicable in real-world video and robotics applications. Our approach combines the advantages of modern methods with the practicality of simpler architectures, aiming to contribute to the field of monocular depth estimation with higher-quality results and increased applicability.

Introduction

In the evolving landscape of computer vision, the estimation of depth from visual input plays a pivotal role in enabling machines to perceive and interact with the three-dimensional world. This report delves into the realm of monocular depth estimation, an area that has garnered significant attention due to its practical applications and the challenges it presents. Unlike stereo vision systems that rely on multiple cameras, monocular depth estimation seeks to infer depth information from a single camera input. This approach not only simplifies the hardware requirements but also broadens the potential applications, ranging from autonomous vehicles to augmented reality and robotics.

Our focus is on developing an efficient monocular depth estimation model that navigates the constraints of computational resources while maintaining high accuracy and reliability. This report presents our methodology, which leverages a streamlined encoder-decoder architecture, complemented by advanced techniques in machine learning. We address the challenges of achieving high-quality depth perception with a single camera, emphasizing the enhancement of depth map quality and boundary precision.

Through a combination of transfer learning, sensor fusion, and strategic data set selection, our approach aims to set a new benchmark in monocular depth estimation, we plan on deploying a single camera based depth estimation model for reactive approached on a 1/10th scale racing car. The subsequent sections will detail our methodology, experiments, results, and the implications of our findings in the broader context of computer vision and its practical applications.

Related Works

Monocular depth estimation, traditionally treated as a regression problem from single RGB images, has seen advancements through CNN methods. Despite progress, issues persist regarding the quality and resolution of depth maps. Our primary focus is on elevating depth map quality and boundary accuracy using conventional neural network architectures.

Recent research explores multi-view stereo reconstruction with CNNs, addressing sub-problems such as image pairs and three consecutive frames. Our specific goal is to advance single-image depth estimation.

Transfer learning has also proven effective in various 3D reconstruction tasks. Our approach heavily relies on transfer learning, leveraging image encoders originally designed for image classification. Importantly, encoders that retain spatial resolution, in combination with skip connections, enhance depth estimation.

Encoder-decoder networks, widely applied in vision

#*****

tasks such as image segmentation, optical flow estimation, and image restoration, have found success in supervised and unsupervised depth estimation.

Modern methods in the field of monocular depth estimation have leveraged the power of Vision Transformers, Attention models, and Recurrent Neural Networks to achieve robust depth estimation with temporal consistency. However, given the time constraints and the scope of our project, we intentionally adopted a simpler encoder and decoder architecture to balance performance with practicality.

Dataset Description: DIODE Dataset

For our project, we plan to use the Dense Indoor Outdoor Dataset, a widely utilized indoor scene dataset in the field of computer vision. This dataset offers: Diverse Indoor Scenes, Depth Maps from RGB-D Sensors, Corresponding RGB Images, Scene Diversity (e.g., bedrooms, kitchens, offices), Human-annotated Depth Information

One limitation of the DIODE dataset is that it does not provide consecutive frames, which is essential for improving the temporal consistency of a model, particularly in video and robotics applications.

The need for a sequence of images is crucial as it allows the model to capture the dynamic changes in a scene over time. In videos, this temporal information helps the model to produce more accurate and consistent depth estimations, especially when objects are in motion or the viewpoint changes. In robotics, understanding and predicting depth across consecutive frames is essential for tasks like autonomous navigation, object tracking, and scene mapping.

Self attained Dataset

To address the limitation of the DIODE, we used Realsense D435i / ZED 2 which includes a sequence of images from a video. There were too many NAn values we obtained from the real sense, which made it difficult to process and train the model. Due to this, we continued with the diode dataset. The Nan values existed over the edges, which would cause problems with the loss function. The data-set can be accessed here: `https://drive.google.com/drive/folders/1RGOImT-y9pZ1VD0D37JJ-ZwN_lhIPfAR?usp=sharing`

Method

Our approach encompasses a monocular depth estimation model tailored for self-driving applications. We initially trained our encoder, but to improve the results, we leveraged pre-trained DenseNet, ResNet, and VGG16 backbones as encoders, We formulated a decoder trained the network on the DIODE dataset. We used a diverse set of loss functions (smoothness, SSIM, L1) and activation functions (ReLU, Leaky ReLU, Sigmoid, Tanh).

We explored various activation functions were, with comprehensive trials on both ReLU and Leaky ReLU. This methodology yielded a robust depth estimation model with applications in 3D point cloud mapping and self-driven car movement direction detection.

0.1. Architecture

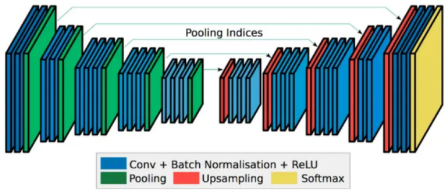

Figure 1. Model Architecture

For the architecture of the network, we took a U-Net model, which has mainly three kinds of blocks.

- Upscale: UpScale block is part of the decoder network, which takes the information from the parallel downscale block, and previous upscale block, and concatenates to give a new output. The features from previous upscale block and upsampled, and increases the resolution of the feature map. It decodes the feature map to the depth image field.
- Downscale: The Downscale box, gives a feature vector in the same resolution, which is then downsampled using max pool, to feed to the next downscale block. It acts as the encoder to create feature maps.
- Bottleneck: The bottleneck layer acts as a buffer between the encoder and the decoder system.

We implemented a total of 4 blocks for downscale and upscale blocks in the model, with the channels of sizes (16,64,128,256,512) with the VGG16 backbone, and (64, 128,256,512,1024) for DenseNet, and ResNet50 model.

0.2. Losses

To train our depth estimation model, we experimented with various loss functions to optimize the model parameters. The following loss functions were employed:

#*****

- Smoothness Loss: Smoothness loss checks for a smooth transition between the depth maps. It is the summed gradient of the prediction image and minimizes it.
- SSIM (Structural Similarity Index): It is the measure of structural similarity between the prediction and the target image.
- L1 loss: We are using L1 loss as the regression loss between the prediction and the target.

We tried various loss weight distributions and finally set the weights as 0.25 for L1 loss, 0.9 for SSIM Loss, and 0.6 for the Smoothness Loss.

0.3. Regression vs Classification

In an attempt to address resolution challenges, we explored an alternative strategy inspired by YOLO/SOLO models, transforming the depth estimation problem into a segmentation task. Utilizing a classification paradigm, we categorized normalized depth values between 0 and 1 into 20 classes. Our model was trained using cross-entropy and dice loss functions, with a nuanced error term scaling of 0.05 for each class.

Despite the conceptual inspiration from YOLO/SOLO, our experimentation on the DenseNet architecture yielded results significantly below our expectations. The performance of the segmentation-based model was notably suboptimal when compared to the outcomes achieved with our original regression-based approach. This led us to conclude that, for our specific application, the regression model better addressed the challenges associated with depth resolution, offering superior accuracy and reliability.

0.4. 1/z

To address the challenge of maintaining resolution in faraway points, we incorporated a depth normalization strategy into our model. The final output of our depth estimation model was transformed using the function $\frac{1}{z}$, where z represents the predicted depth. This normalization technique scales the depth values to a range between 0 and 1. Consequently, objects nearby exhibit depth values approaching 1, while those in the distance approach 0. By implementing this approach, we aimed to enhance the model's ability to capture fine details in objects situated at varying distances, mitigating resolution degradation in faraway regions and contributing to the overall accuracy of the depth predictions.

Although, the results didn't turn out to be quite as what we expected.

Results

0.5. Inital Results

Here are the revised results obtained from our initial models, which were configured without pretrained backbones and employed Tanh and Sigmoid activations in the final layer. Additionally, the classification model's outcomes are presented. The encoder, not having been trained on a diverse set of images and influenced by a significant class imbalance—wherein depth values are predominantly larger in most regions—exhibited a bias towards estimating greater depth values.

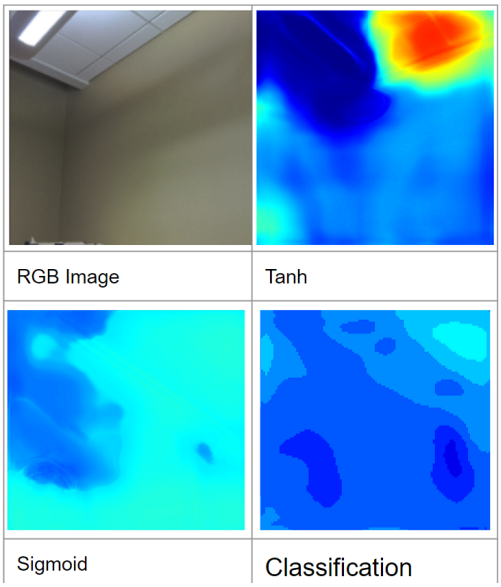

Figure 2. Training Losses

#*****

0.6. Results from Model with Pretrained Backbone

Shown Below are the results from the model trained on VGG-16 Backbone on the train section of DIODE Dataset (81GB)

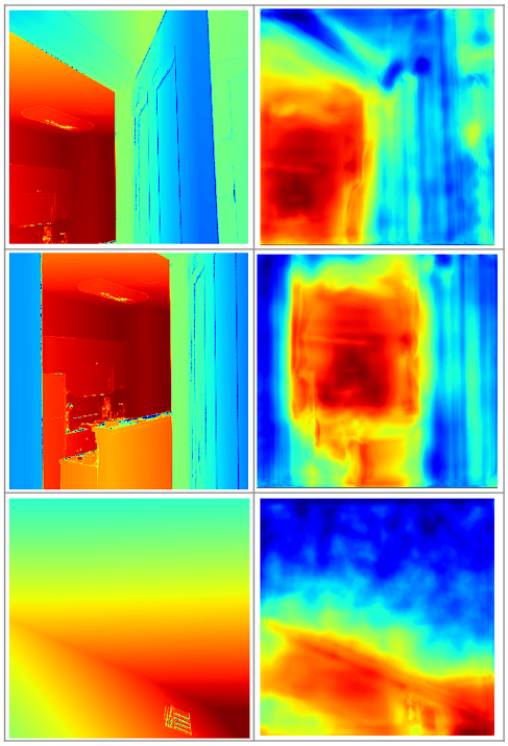

Figure 3. Ground Truth vs Model Predictions

0.7. Loss

Figure 4. Training SSIM,Edge, L1 of Final Model

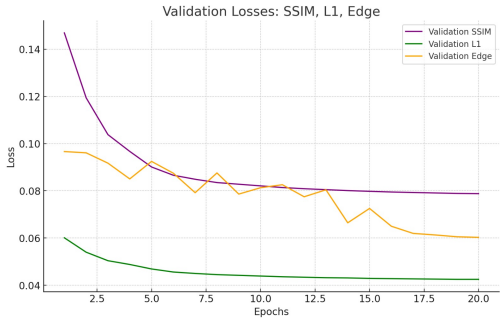

Figure 5. Validation SSIM,Edge, L1 of Final Model

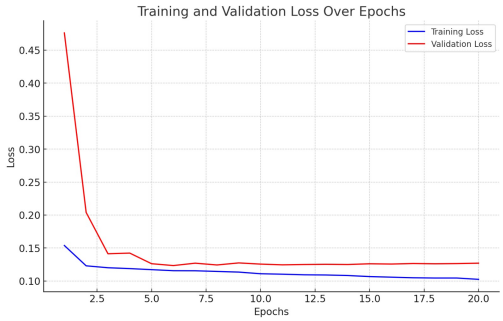

Figure 6. Total Training and Validation Loss

Recovering Absolute Depths

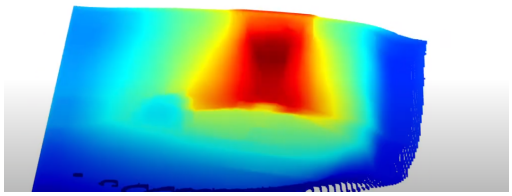

Figure 7. Point Cloud of Depth

0.8. Sensor Design

The sensor design involves incorporating a Time-of-Flight (ToF) sensor array and a camera for depth perception. The ToF sensor array captures range measurements, while the camera provides visual data. The relative depths are inferred using the monocular depth estimation model. To recover absolute depths, the depth map is fused with the direct range measurements obtained from the ToF sensor array. This fusion process combines the strengths of both methods, resulting in more reliable and precise depth measurements.

#*****

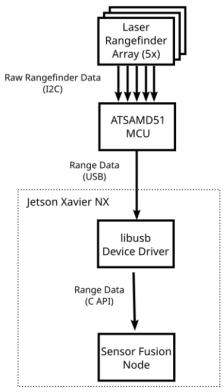

Figure 8. Sensor Pipeline

The sensor design incorporates the VL53L4CX, a cost-effective solution known for its excellent depth accuracy. However, it comes with a fixed 18-degree field of view (FOV). To cover the full FOV of the camera, an array of rangefinders is utilized. The sensor data is communicated through a chain of components. The sensor communicates with a SAMD51 microcontroller via I2C protocol. The SAMD51 then transfers the data to a computer through a USB connection using the libusb library.

Figure 9. Sensor and Camera Mount

To ensure proper alignment and coverage, a 3D-printed mount is employed. The mount positions the rangefinders at fixed 18-degree increments, matching the FOV of the VL53L4CX. This arrangement guarantees comprehensive depth measurements across the entire FOV of the camera.By combining the low-cost and high-accuracy VL53L4CX sensor with an array of rangefinders, efficient data transmission via I2C and USB, and a well-designed 3D-printed mount, this sensor design provides a practical solution for achieving accurate depth perception within a fixed 18-degree FOV.

0.9. Recovering Scale from Depth Maps with Fusion

The process of recovering absolute depths involves several steps. Firstly, intensity peaks are correlated and linear regression is performed to estimate the relationship between these peaks and actual depth values. Depth maps are then overlapped and analyzed using histogram techniques, enabling the identification of common depth regions and refining depth estimates. Scaling factors and biases are obtained to align depth map predictions with absolute inverse depth measurements, ensuring accurate depth representation.

In the next step, depth values are adjusted based on the obtained factors and biases, aligning them with the absolute depth measurements. A sensor fusion algorithm is then applied, integrating depth information from multiple sources such as cameras and range finders. This fusion algorithm combines the adjusted depth values to create a comprehensive and accurate depth representation. Together, these steps contribute to the recovery of absolute depths, providing a robust and reliable understanding of the scene.

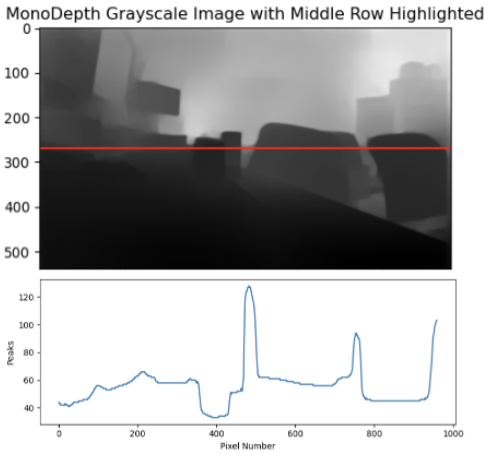

Figure 10. Histogram Analysis of Depth Map

0.10. Emulating LiDAR Like 2D scans

After recovering the Scaling Factor, we proceed to apply it uniformly across all the pixels in the image. This step is followed by extracting the central row of the image, creating a simulation reminiscent of a LiDAR scan. The image presented below illustrates the emulated LiDAR scan, achieved through the integration of our monocular depth estimation model and sensor fusion techniques. This demonstration highlights the efficacy of our trained model in replicating LiDAR-like point cloud scans using just a single camera, offering a cost-effective alternative to traditional methods.

#*****

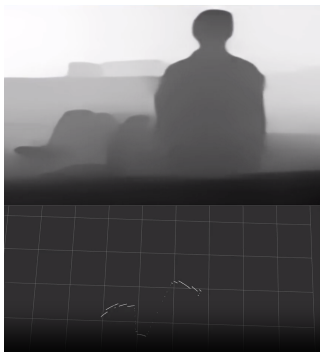

Figure 11. Emulated LIDAR Scan from Depth Map

Future Improvements

We can implement ViT(Vision Transformers) ViT applies the transformer architecture, primarily used in NLP, to image processing tasks. It divides images into patches and processes these sequentially through attention mechanisms. Advantages: Capable of capturing long-range dependencies in the image data. Often outperforms traditional CNNs in large-scale image recognition tasks.

Application in Depth Estimation: Potential for improved contextual understanding and depth perception over large areas. Still in the exploratory stages for depth estimation but shows promising results.

0.11. Concept:

Self-supervised learning uses the data itself as a supervisory signal, eliminating the need for labelled datasets. Depth estimation often involves reconstructing images or predicting adjacent frames in a sequence.

0.12. Techniques:

- Monocular video-based training, where depth is inferred by predicting future frames.
- Use of geometric constraints or consistency losses to guide learning.

0.13. Benefits:

- Removes dependency on large, labeled datasets.
- Can be more adaptable to different environments and lighting conditions.

0.14. Current Challenges:

- Often requires more complex training strategies.
- May struggle with absolute depth accuracy compared to supervised methods.

Conclusion

In our pursuit of an effective monocular depth estimation model for self-driving applications, we conducted a thorough exploration of diverse strategies and architectures. We experimented with a classification-based approach inspired by YOLO/SOLO models, utilizing cross-entropy and dice loss functions for 20 depth classes. Despite our efforts, this methodology yielded sub-optimal results, underscoring the nuanced nature of our depth prediction task.

Additionally, we addressed resolution challenges by normalizing depth values using $\frac{1}{z}$, aiming to enhance performance in distant regions. However, our comprehensive trials across various backbones—ResNet, DenseNet, and VGG16—revealed that VGG16 consistently delivered the best results. We postulate that the skip connections and concatenations inherent in ResNet and DenseNet architectures might compromise spatial features crucial for our depth estimation task.

In summary, our exploration emphasized the efficacy of a regression-based approach, particularly with VGG16 as the backbone. This methodology not only exhibited superior accuracy but also showcased the importance of preserving spatial features for the nuanced task of monocular depth estimation in self-driving scenarios.

References

- Monocular Depth in the Real World - Toyota Research Institute
- Monocular Depth Estimation - Keras
- MiDaSv3.1- A Model Zoo for Robust Monocular Relative Depth Estimation - Intel Labs
- NYU Depth Dataset- v2
- Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-shot Cross-dataset Transfer