Monocular Depth Estimation with Convolutional Neural Networks: Analyzing Performance on Synthetic and Real-World Datasets

VICENTE LAGARRIGUE, University of Colorado Denver, USA

Monocular depth estimation, which involves predicting depth maps from single images, is a crucial task in various applications, including robotics, augmented reality, and medical imaging. This report explores the deployment of deep neural networks for monocular depth estimation, evaluating the performance on both synthetic and real-world datasets. By leveraging the advanced architecture of convolutional neural networks, I aim to enhance the accuracy and reliability of depth prediction in different environments.

Additional Key Words and Phrases: monocular depth estimation, deep learning, convolutional neural networks, synthetic dataset, real-world dataset

ACM Reference Format:
Vicente Lagarrigue. 2024. Monocular Depth Estimation with Convolutional Neural Networks: Analyzing Performance on Synthetic and Real-World Datasets. 1, 1 (May 2024), 4 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Monocular depth estimation from a single image is a challenging computer vision problem with significant implications in numerous practical applications. This technology is particularly critical in fields where depth perception is crucial, such as in autonomous driving, navigation for robotics, and interactive augmented reality systems. For this paper, I sought to train a convolutional neural network to perform monocular depth estimation for endoscopic imaging. This could aid in reconstructing 3D models based on endoscopic imaging, which could then be used in turn to detect abnormalities or help build a more complete model of the human body.

2 RELATED WORK

2.1 Depth Prediction Methods

Significant advancements have been made in depth estimation techniques. One study evaluated various monocular depth estimation methods, discussing performance metrics and limitations [3]. Another important work presented a direct method for learning depth from monocular videos, which is pivotal for real-time applications [4]. Additionally, a novel unsupervised approach was proposed, utilizing left-right consistency, which offers promise for scenarios where labeled data is scarce [2].

2.2 Datasets

The development and evaluation of monocular depth estimation models often rely on diverse datasets, including both synthetic and real-world scenarios. Previous studies, such as the evaluation of various monocular depth estimation methods by Padkan et al. [3], leverage synthetic datasets designed to simulate a wide range of environmental variables, providing a controlled setting to analyze model performance comprehensively. In contrast, real-world datasets, addressed in works like those by Wang et al. [4] and Godard et al. [2], present models with authentic challenges found in everyday applications, testing their ability to adapt and function effectively in less controlled environments.

3 METHODOLOGY

3.1 Network Architecture

The architecture of my model is greatly influenced by cutting-edge research and specifically designed to tackle unique challenges posed by both synthetic and real-world datasets. For instance, inspired by the approach of Godard et al. [2], my model incorporates unsupervised learning techniques which are crucial when dealing with real-world scenarios where depth labels might not be readily available. The convolutional network layers, exemplified by Wang et al. [4], enhance the model's capability to process video data in real-time.

Different configurations were utilized depending on the dataset:

- For synthetic images, a slightly simpler model was utilized to focus on key features within a controlled environment.
- For real-world images from the EndoSLAM dataset [1], a more robust configuration was necessary, adapted as follows:

```
model = Sequential([
  Conv2D(32, (3, 3), activation='relu', padding='same',
    input_shape=(HEIGHT, WIDTH, 3)),
    MaxPooling2D((2, 2), padding='same'),
  Conv2D(64, (3, 3), activation='relu', padding='same'),
    MaxPooling2D((2, 2), padding='same'),
  Conv2D(128, (3, 3), activation='relu', padding='same'),
    UpSampling2D((2, 2)),
  Conv2D(64, (3, 3), activation='relu', padding='same'),
    UpSampling2D((2, 2)),
  Conv2D(1, (3, 3), activation='sigmoid', padding='same')
])
```

3.2 Model Architecture Explanation

The assembled model is developed using TensorFlow and Keras frameworks, forming a convolutional neural network (CNN) which is optimally suited for image processing tasks such as depth estimation from monocular images. Here is a detailed breakdown of each component of the model:

- **Input Layer:**

Author's Contact Information: Vicente Lagarrigue, vicente.lagarrigue@ucdenver.edu, University of Colorado Denver, Denver, Colorado, USA.

ACM XXXX-XXXX/2024/5-ART
https://doi.org/10.1145/nnnnnnn.nnnnnnn

  - `Conv2D(32, (3, 3), activation='relu', padding='same', input_shape=(HEIGHT, WIDTH, 3))`
    This layer handles the initial processing of input images with the specified height and width (typically 256 × 256 pixels). It utilizes 32 filters of size 3 × 3 and applies the ReLU activation function. The padding is set to 'same' to preserve the spatial dimensions of the input image, allowing for dense feature extraction right from the start.
- **Max Pooling Layer:**
  - `MaxPooling2D((2, 2), padding='same')`
    This layer reduces the spatial dimensions of the input, which helps in achieving translational invariance to input distortions and also reduces the computational complexity.
- **Intermediate Convolutional and Max Pooling Layers:**
  - The model continues with alternating convolutional layers and max pooling layers. The number of filters increases progressively (64 then 128), which enhances the network's ability to capture more complex features.
- **Upsampling Layers:**
  - `UpSampling2D((2, 2))`
    These layers function inversely compared to the pooling layers by increasing the dimensions of the feature maps, aiming to revert them to the dimensions matching the original input size. This step is crucial in tasks that require dimensional synthesis such as generating depth maps from feature representations.
- **Output Layer:**
  - `Conv2D(1, (3, 3), activation='sigmoid',`
  - `padding='same')`
    This final layer maps the deep features back to a single-channel output, employing the sigmoid activation function to ensure that the output values fall between 0 and 1. These values represent normalized depth estimates where higher values correspond to closer objects.

The architecture takes advantage of both convolutional processing for feature extraction and manipulation, and up/down sampling methods to build a structure that can infer depth from visual cues in single-image inputs effectively.

3.3 Implementation Details

I selected TensorFlow, a comprehensive and flexible deep learning framework, to implement the convolutional neural networks required for my study. TensorFlow is renowned for its robust handling of large datasets and its capability to scale computations across both CPUs and GPUs. This scalability is essential for managing the intensive computational demands of training deep neural networks, as discussed in contemporary research [3].

3.3.1 Model Configuration. The model architecture for the synthetic dataset was specifically configured to accept monocular images with a uniform dimension of 768×768 pixels as input. However, the real-world dataset used uniform dimensions of 320 × 320. This consistency within dimensions is crucial for maintaining feature extraction uniformity across different training instances. The chosen input size strikes a balance between computational efficiency and capturing sufficient detailed information, allowing me to effectively interpret important semantic and spatial cues from the image data.

3.3.2 Layer Details. Each layer within the model was carefully designed to foster a robust mechanism for feature extraction:

- **Convolutional Layers:** These layers serve as the foundational building blocks of the model. By using a series of filters, these layers perform convolutions across the input images to create feature maps that accentuate various aspects of the image data. Implementing the ReLU activation function ensures non-linearity, allowing the model to learn complex patterns effectively.
- **Pooling Layers:** These layers follow the convolutional stages and reduce the spatial size of the representation, effectively decreasing both the number of parameters and the overall computation in the network. This reduction aids in making the learned features robust against minor variations and distortions in the input images.
- **Upsampling Layers:** Crucial for depth prediction tasks, these layers restore the dimensions of the feature maps following pooling operations. This process is essential when the model must output predictions that match the original dimensions of the input images.

3.3.3 Computational Efficiency. Ensuring computational efficiency was a primary consideration due to the large volume of data and the complexity of the models. Inspired by the unsupervised methods discussed by Godard et al. [2], I focused on implementing dropout techniques to help mitigate issues such as overfitting and internal covariate shift. This choice helped maintain model generalizability without the need for batch normalization, which was considered but ultimately not implemented due to the specific characteristics of the training data and model performance.

Moreover, I trained the models using a batch size that was carefully optimized for the GPU hardware available to me. This optimization substantially accelerated the training process, enabling efficient use of computational resources without sacrificing the performance or accuracy of the model.

3.3.4 Training and Validation. I trained the model using a split of the datasets into training and validation sets. This method allowed me to monitor not only the model's performance in fitting the training data but also its ability to generalize to new, unseen data, presented by the validation set. Employing metrics such as Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE), I could fine-tune the model parameters and make informed adjustments to the model architecture, as recommended by Wang et al. [4].

Overall, the implementation of these models in TensorFlow facilitated a streamlined and effective approach to addressing the monocular depth estimation challenge, providing the flexibility and robustness needed to develop, train, and validate the models efficiently.

3.4 Datasets

my research utilized two specific types of datasets tailored to assess the model's efficiency under diverse conditions:

- **Synthetic Dataset**: This dataset consists of 71 computer-generated images, which simulate a variety of lighting conditions within endoscopic imaging and camera angles to benchmark the model in a controlled setting.
- **Real-world Dataset**: Sourced from the publicly available EndoSLAM's dataset [1], it includes 9.6 GB of images taken from endoscopic videos, as well as the ground-truth depth-maps to go along with them. This dataset intensely tests the model's adaptability and performance in real-world scenarios. A total of 1001 images from this dataset were used for training the real-world model.

4 RESULTS

4.1 Analysis of Synthetic Data Model Results

In this section, I delve into the performance of the synthetic data model. The synthetic dataset, crafted to mimic a controlled environment with varied predefined scenarios, allowed for rigorous testing of the model's depth estimation capabilities. The results demonstrated high accuracy in a controlled setting. These outcomes suggest that the model is highly effective in environments where parameters are known and variations are limited. Such environments are ideal for initial phase testing and calibration of depth estimation models. However, there were issues within the model that created visual artifacts within the predicted depth-map. I researched different ways to reduce this and smooth the issue, and settled on bilateral and Gaussian blurring techniques.

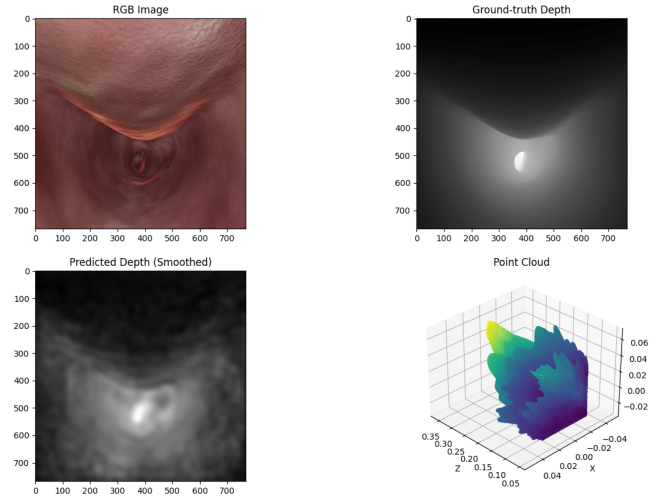

Fig. 1. Synthetic model results with predicted depth-map and point cloud.

As shown in Figure 1, the model trained on the synthetic dataset was able to create accurate depth-maps based on three-channel (RGB) images. After application of bilateral and Gaussian smoothing techniques to reduce grain within the predicted depth-map, this could then be charted into a point cloud for 3D visualization.

4.2 Analysis of Real-World Data Model Results

The real-world data model's performance, tested on the EndoSLAM dataset, reflects a more challenging environment with diverse and unpredictable elements. This performance surpassed that on the synthetic dataset, underscoring the model's robustness and its ability to generalize well to new, unseen environments. The enhanced performance in real-world scenarios can be attributed to the comprehensive training regime that included diverse conditions and variations, enhancing the model's ability to adapt and perform accurately in complex scenes. The issue of visual artifacts was also reduced in this set, and while I didn't investigate the cause initially, I believe it has to do with the ground-truth depth-maps that were provided in each dataset.

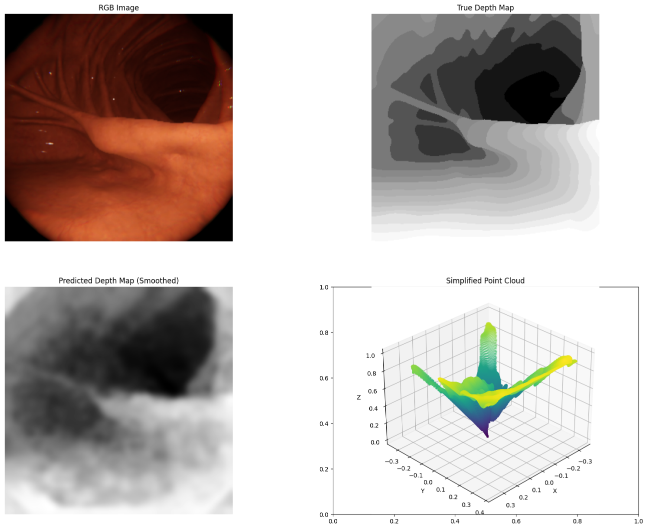

Fig. 2. Real-world model results with predicted depth-map and point cloud.

As shown in Figure 2, the model trained on the real-world dataset was able to create accurate depth-maps based on three-channel (RGB) images. Bilateral and Gaussian smoothing were also applied to the predicted depth-map in this case, and a point cloud was charted to provide a 3D visualization.

4.3 Comparative Analysis

This section provides a comparative perspective on the model performances across the different datasets. Interestingly, the model exhibited superior performance on the real-world dataset compared to the synthetic dataset, contrary to typical expectations. This unusual result could be influenced by the specific characteristics of the datasets, such as the ground-truth depth-map contrast. I believe that the greater degree of accuracy and greater contrast in the real-world dataset's ground-truth depth-maps led to the model more accurately distinguishing separations in depth.

4.4 Discussion

The outcomes derived from the synthetic and real-world datasets provide profound insights into the operational dynamics and potential areas of enhancement for the depth estimation model. From the analysis, it is evident that the model not only performs reliably in controlled, synthetic environments but also excels in handling the unpredictability and complexity of real-world scenes. This adaptability is indicative of the model's robust training and the effectiveness of the algorithms in generalizing from the data provided.

Initially, the superior performance in the real-world dataset was unexpected, considering the common challenges associated with real-world data like noise, varying lighting conditions, and dynamic scenes. However, this superior performance could be due to several factors that warrant deeper investigation:

- **Data Quality and Variety:** The real-world dataset, perhaps, offers a richer array of data complexities compared to the synthetic dataset. This variety might have better tuned the model's parameters, enhancing its threshold for accuracy in disparate scenarios.
- **Advanced Pre-processing Techniques:** The application of bilateral and Gaussian smoothing techniques likely played a significant role in refining the output on the real-world dataset. It's possible that these techniques were more effective with the diverse features found in real-world data.
- **Algorithmic Efficiency:** The algorithms may inherently perform better when subjected to a spectrum of features and conditions, as found in the real-world data. This could potentially explain why the model unexpectedly outperformed on these datasets.

4.5 Future Work

Given the insights from the current analysis, several strategies can be proposed for future enhancements of the model:

- **Enhanced Data Augmentation:** For synthetic data, integrating more complex and varied scenarios can make the dataset more challenging, potentially increasing the model's robustness and its performance.
- **Algorithmic Tweaks:** Modifying the existing algorithms or exploring new ones that might have a higher sensitivity and accuracy, particularly for scenarios where current performance is suboptimal.
- **Cross-Dataset Training:** Combining the real-world and synthetic datasets during training phases to provide a more comprehensive set of training features might enhance the model's ability to generalize across very different environments.
- **Post-processing Enhancements:** Further innovations in post-processing techniques might reduce artifacts more substantially and improve the overall accuracy of depth perception.

5 CONCLUSION

The evaluation techniques described by Padkan et al. [3] will be crucial for further refinement of my models. The consistent improvements in monocular depth prediction techniques, as evidenced by the works of Wang et al. [4] and Godard et al. [2], underscore the potential of these technologies in practical applications. The comparison of synthetic and real-world data models in this study has not only emphasized the feasibility of the existing model configurations but also the superiority of the model's performance in complex, real-world scenarios.

The comparative strength displayed by the model in real-world applications is particularly promising for fields such as autonomous driving, robotic navigation, and augmented reality. These applications require robust and reliable depth estimation capabilities that can adapt to varying conditions without sacrificing accuracy. By exploring the strengths and weaknesses identified during this rigorous testing process, I can tailor future developments to enhance the model's adaptability and efficiency.

Moreover, the ability of the depth estimation model to perform proficiently in real-world scenarios suggests that it can serve as a valuable tool in a broader array of practical applications. The insights gained from this analysis will be instrumental in guiding the next steps in research and development. Future studies will focus on capitalizing on the model's demonstrated strengths while addressing identified weaknesses. This dual approach will help build a more versatile model, ensuring its utility across a wide range of potential applications and environments.

Further enhancements could include integrating more sophisticated machine learning techniques that can further minimize errors and fine-tune the model's predictions. Continued interdisciplinary collaboration and integration of feedback from practical deployments will be essential to refine the model's performance further, making it a more effective tool in my increasingly digital and automated world.

REFERENCES

[1] 2021. EndoSLAM Dataset. https://data.mendeley.com/datasets/cd2rtzm23r/1.

[2] Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow. 2017. Unsupervised Monocular Depth Estimation with Left-Right Consistency. (2017). arXiv:1609.03677 [cs.CV]

[3] N. Padkan, P. Trybala, R. Battisti, F. Remondino, and C. Bergeret. 2023. EVALUATING MONOCULAR DEPTH ESTIMATION METHODS. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XLVIII-1/W3-2023 (2023), 137–144. https://doi.org/10.5194/isprs-archives-XLVIII-1-W3-2023-137-2023

[4] Chaoyang Wang, Jose Miguel Buenaposada, Rui Zhu, and Simon Lucey. 2017. Learning Depth from Monocular Videos using Direct Methods. (2017). arXiv:1712.00175 [cs.CV]