SYNCHRONIZED LEARNING OF DEPTH ESTIMATION AND SEMANTIC SEGMENTATION

Prateik Sinha
Department of Mathematics
University of California, Los Angeles
Los Angeles, CA 90095, USA

Shivam Kumar Panda
Department of Mechanical and Aerospace Engineering
University of California, Los Angeles
Los Angeles, CA 90095, USA

ABSTRACT

Jointly learning the tasks of depth estimation and semantic segmentation has shown significant gains in performance for CNN based models. We want to show this technique can be extended to other architectures. We propose a new method to jointly learn these tasks and show that this method can be used to improve the performance of state-of-the-art models in both these domains. See:
`https://github.com/Prateik-11/semantic-depth-transformer`

1 INTRODUCTION

We all have an intuitive understanding that depth perception and semantic understanding of a scene are intrinsically linked. If we close one eye, we have no depth perception but due to our ability to recognize objects and their sizes from memory (among other visual cues) we can still accurately estimate the distance of objects.

It stands to reason that neural networks trying to learn either of these tasks could see an increase in performance if they already knew how to do the other. Several papers have verified this hypothesis by documenting the increase in performance achieved by transfer learning between these tasks on CNN based architectures. In this paper we apply these techniques to modern transformer models and see if we can leverage this technique on newer architectures. For this purpose, we propose a method to combine the architectures of these models (Synchronized Framework) as well as a contrastive loss function to ensure consistency between the models.

We test our technique on a UNet as a proof of concept, then we apply our method to current state-of-the-art methods to see if leveraging joint learning of these tasks can improve their performance and push the current limit of depth estimation and semantic segmentation.

This is also useful because semantic segmentation data is very limited. Since each pixel needs to be labelled by hand as belonging to one of n different classes, this is very expensive to generate. Depth estimation, on the other hand, is very easy to find data for if self-supervised techniques are used. Labelled depth maps are no longer needed, only stereo images or sequences of images (in conjunction with a pose estimation model) are sufficient for training. Hence being able to reduce the required training data using a pre-trained depth model would prove useful.

2 PREVIOUS WORK

2.1 DEPTH ESTIMATION VS CLASSIFICATION AS PRE-TRAINING FOR SEMANTIC SEGMENTATION (2023)

(5) This paper states that semantic segmentation data is very labor-intersive to create, and thus models need to be pre-trained on another task first to reduce the amount of training data needed. Semantic segmentation networks are typically pre-trained for classification tasks on datasets such as ImageNet. This paper shows that pre-training using depth estimation improves the performance of downstream semantic segmentation compared to classification, on average by 5.8% mIoU and 5.2% pixel accuracy. Even on reducing the number of fine tuning data points from 128 down to 8, depth estimation consistently outperforms ImageNet pretraining. This proves our premise that the

Figure 1: Example of RGB Image used as input from Virtual KITTI Dataset

Figure 2: Example of ground truth Depth Map from Virtual KITTI Dataset

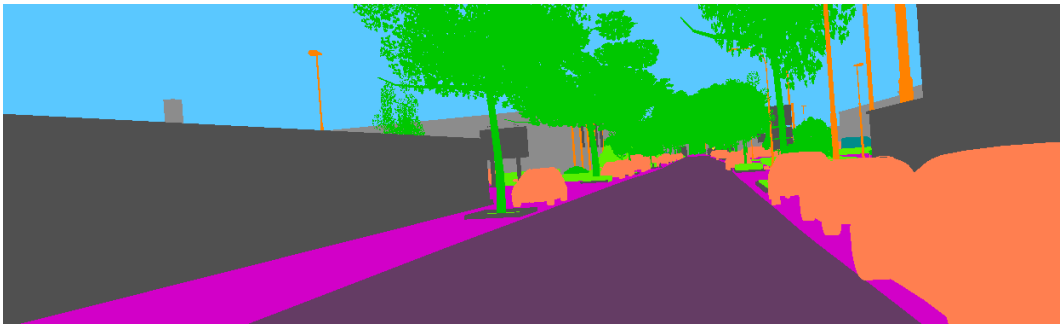

Figure 3: Example of ground truth Semantic labels from Virtual KITTI Dataset

link between the two tasks is significant and there are performance gains to be found by learning them jointly.

The point of this paper is to prove the viability of depth estimation as a valid pre-training task. In practice one can use it in combination with image classification to get the best results. Pre-training on a depth network after ImageNet initialization yields better depth estimates than pre-training on a depth network trained from scratch

2.2 Improving Semi-Supervised and Domain-Adaptive Semantic Segmentation with Self-Supervised Depth Estimation

(4)

This paper attempts to fix the lack of semantic segmentation data by enhancing the model using self-supervised depth estimation models. It provides methods to:

1. Automatically select the most useful samples to be annotated for semantic segmentation based on the correlation of sample diversity and difficulty between SDE and semantic segmentation.
2. Augment data by mixing images and labels using the geometry of the scene.
3. Use transfer learning to transfer knowledge learned during SDE over to semantic segmentation
4. Use additional labeled synthetic data with cross domain depth mix and matching geometry sampling to align synthetic and real data

Each of these 4 leads to a significant performance gain when tested on the CityScapes dataset.

1. Achieves 92% of the full accuracy with 1/30th of the data
2. Achieves 97% with additional data from the GTA dataset

While this paper provides several methods to alleviate the lack of semantic segmentation data and achieve near full accuracy with only a fraction of the dataset, it does not seem to able to improve beyond the original accuracy achieved with all the data intact.

2.3 Learning depth via leveraging semantics-self supervised monocular depth estimation with both implicit and explicit semantic guidance (2023)

(6)

Conversely to the previous two works, this paper attempts to use semantic data to augment and improve the performance of a depth estimation model rather than the other way around. It does so in two ways: implicitly and explicitly.

1. Explicit method: They propose a semantic-guided ranking loss to explicitly constrain the estimated depth maps to be consistent with real scene contextual properties. This involves randomly sampling points from the output of the model and checking the semantic classes they belong to. If they are in the same class the model with enforce their similarity in the loss function and vice versa. They also use the loss function to incentivize edges in the depth map near where there are edges in the semantic labels.
2. Implicit method: a Semantic-aware Spatial Feature Alignment (SSFA) scheme to align implicit semantic features with depth features for scene-aware depth prediction. This spatially aligns depth and semantic features via constraining depth distribution to be consistent inside the same category area, while to be different across other categories.

2.4 Extra Multi-Task Learning for Dense Prediction Tasks - A Survey

(8)

This paper provides a comprehensive overview of different architectures and optimization methods used to train neural networks on multiple tasks. This overview was useful in deciding upon the

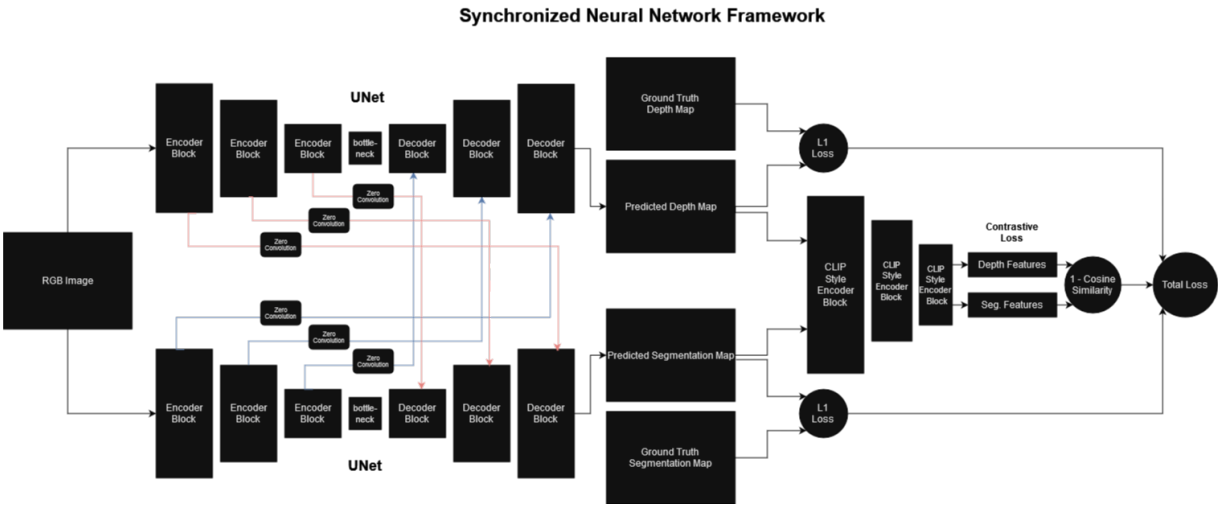

Figure 4: An overview of the framework we propose

architecture we finally chose to combine depth estimation and semantic segmentation in our own model.

2.5 MiDaS v3.1 – A Model Zoo for Robust Monocular Relative Depth Estimation

(1)

MiDaS v3.1 is the current SoTA for depth estimation. It provides a suite of models with varying sizes. All of these are transformer based architectures as of the most recent version. Since many of the papers combining depth and semantics were only tested on CNNs it will be worthwhile to see how their interaction is affected by the different architecture and the self-attention layer.

3 Methods

3.1 Contrastive Loss

Contrastive learning is an approach that maps training data to a latent space such that the cosine similarity between latent representations of similar or matching data points is minimized and that between disparate data points is maximized. As such, once the encoder is trained the similarity between latent representations can be used as a metric to measure the disparity of the data points. We train such a model to minimize the distance between depth map and segmentation map of the same image and maximize the distance when both are derived from different images. We add this similarity score to the loss function of both depth estimation and semantic segmentation models so they both learn consistent representations of the source image in their respective latent spaces.

We train our contrastive learning model in the style of OpenAI's CLIP (7) using a shared image encoder based on the Vision Transformer (2) encoder used in CLIP.

While CLIP was originally intended for comparing pictures with text, we use it for comparing the depth and segmentation maps. Thus, we have two image encoders instead of and image and a text encoder.

The dataset we use is VKITTI since it is one of the few datasets that has depth maps and semantic maps of the same source image and is large enough to train reliably with. We also cannot use an existing model to generate the training data since we intend to use this model to improve the current SoTA.

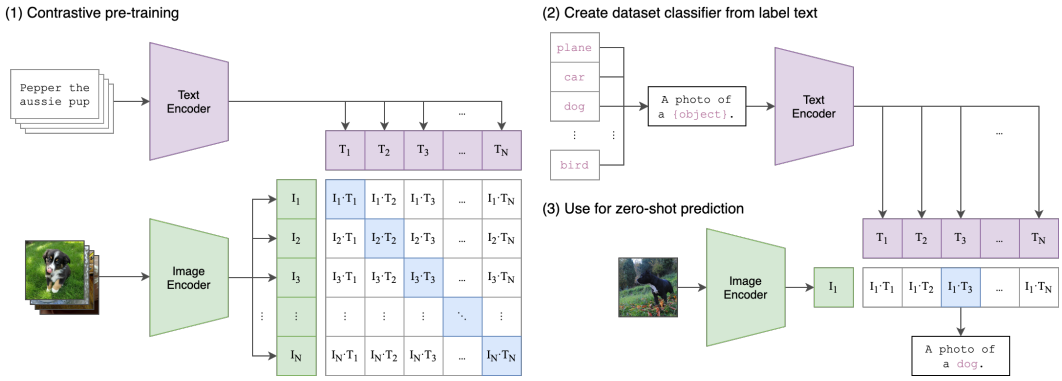

Figure 5: CLIP

3.2 Synchronized Framework

The previous works detailed several different ways to combine the models for depth estimation and semantic segmentation. These include approaches such as mixing and matching the encoder and decoder, pre-training on one task then fine-tuning end-to-end on the other, freezing the encoder after learning one task then fine tuning, and so on. The central conflict which makes choosing the architecture difficult is that we want to share as many layers as possible to leverage the benefits of jointly learning both tasks, but we also want each task to have its own exclusive layers so the model is able to specialize and learn task-specific optimizations rather than just being mediocre at both.

The approach we settled on is inspired by ControlNet (9). This auxillary model is used to guide the image generation process in diffusion models, and to ensure it does not end up hurting the performance, a network is learnt in parallel which adds the outputs of its own encoders through residual connections. Each of these residual connections, however, first passes through a 1x1 convolution that is initialized with weights and biases set to 0. This means that the augmentation network will start off by having no effect on the original model. It will only have an effect if during the training process these weights change due to backpropagation, which will only happen if doing so would reduce the loss. Therefore the addition of the zero convolution makes it so the additional guidance will never hurt and only help the model.

The zero convolution also introduces the phenomena of sudden convergence, which needs to be kept in mind while training. Essentially the guidance will have no effect on the network till the zero convolution is undone, and when it is the loss will drop drastically all at once. This is unlike the incremental reduction of loss throughout the training process which is normally seen.

We can apply this principle to jointly train the depth and semantic models. We use the zero masked residual connections to add the outputs of the encoders of each model to the decoders of the other model. Hence when the model is reconstructing the output from the bottleneck/latent space it is doing so not only with the output of the encoder but also with the encoder that is learning the other task

4 Results

To test whether our design choices work, we train two networks for these tasks. One with our proposed changes and one without them. As a simple baseline to prove our concept, each of these networks is a UNet architecture. The network contains augmentations from the DDPM paper (3), such as attention layers after the convolutional blocks.

While the model worked well on our proof of concept, we need to verify if the model is able to improve the performance of the current state of the art techniques for depth estimation and semantic segmentation.

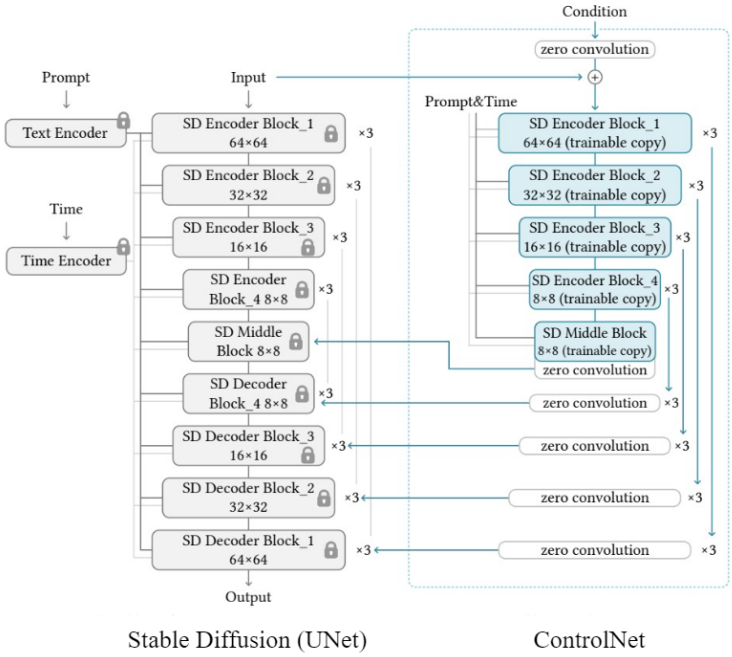

Figure 6: ControlNet Architecture

The current SoTA for depth estimation is the MiDAS model (1). The current SoTA for semantic segmentation is not as clearly defined since each dataset will have its own class of objects that it is trying to predict, hence the task is not as uniform as depth prediction.

Improving the performance of these models using the changes we proposed is our next target.

5 Discussion and Future Work

This research is still a work in progress; further testing need to be performed as outlined in the results section previously. The next steps to take are:

1. Finish training the UNet architectures using our framework with different backbones (ConvNet, ConvNeXtm, ViT, etc) and compare their performances.
2. Replace the UNets with SoTA models for both tasks and try to push their performance beyond their current limit.
3. Repeat this procedure with different datasets to verify that the our method is generalizable.
4. Perform ablation studies to discover which components of our architecture are contributing to the improvements and in what capacity.
5. Experiment with using techniques such as self supervised learning and verify thier compatibility with our framework.

As for further improvements on the sections already complete, there are still more avenues that can be explored:

5.1 Training Methods

The original use of zero convolutions was motivated by the fact that the guidance network was randomly initialized while the central network was already trained, thus the guidance would only hinder the output until it reached some sort of convergence. In this case we train all the networks from scratch all at once, but given the capability of zero convolution we may see better results if we first individually one or both of the networks before we begin the synchronized training procedure

5.2 Additional Datasets

The contrastive network and the UNets were both trained on Virtual KITTI. We may see interesting results if we test or train these models on other datasets. Virtual KITTI, while extremely useful is ultimately still simulated data and may be introducing artifacts or irregularities from real world data that are going unnoticed.

References

[1] Reiner Birkl, Diana Wofk, and Matthias Müller. Midas v3.1 – a model zoo for robust monocular relative depth estimation, 2023.

[2] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.

[3] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020.

[4] Lukas Hoyer, Dengxin Dai, Qin Wang, Yuhua Chen, and Luc Van Gool. Improving semi-supervised and domain-adaptive semantic segmentation with self-supervised depth estimation. *Int. J. Comput. Vision*, 131(8):2070–2096, may 2023.

[5] Dong Lao, A. Wong, and Stefano Soatto. Depth estimation vs classification as pre-training for semantic segmentation. 2022.

[6] Rui Li, Xiantuo He, Danna Xue, Shaolin Su, Qing Mao, Yu Zhu, Jinqiu Sun, and Yanning Zhang. Learning depth via leveraging semantics: Self-supervised monocular depth estimation with both implicit and explicit semantic guidance, 2021.

[7] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021.

[8] Simon Vandenhende, Stamatios Georgoulis, Wouter Van Gansbeke, Marc Proesmans, Dengxin Dai, and Luc Van Gool. Multi-task learning for dense prediction tasks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, page 1–1, 2021.

[9] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023.