Group 31
Dense Depth Estimation Using Transformers

Linus Grasel, Siyuan Zhu
Department of Electrical and Computer Engineering
University of California, San Diego
La Jolla, CA 92093

***Abstract*—Deep learning has been rapidly evolving, showing major advancements at achieving outstanding performance on image processing tasks, such as image classification, objective detection and semantic segmentation etc.. Among these recent developments, it has shown that the pixel-level depth map can be recovered from a single image in an end-to-end manner. Over time, a variety of deep neural networks have manifested their effectiveness to address the monocular depth estimation, i.e. the task of estimating scene depth using a single image, such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), variational auto-encoders (VAEs), generative adversarial networks (GANs) and most recently Vision Transformers (ViT). In this paper we will focus on the latter, Vision Transformers (ViT), more specifically the state-of-the-art model known as AdaBins[1].**

I. Introduction

Estimating dense depth information is essential and helpful for many computer vision tasks such as object detection, semantic segmentation, navigation, etc. and thus it has become very important for many autonomous systems to help perceive their state in the world. However, traditional depth estimation methods, like structure from motion (SfM)[3] and stereo vision matching, are built on feature correspondences of multiple viewpoints. Meanwhile, the predicted depth maps are sparse. More recent sensor-based solutions such as RGB-D cameras and LIDAR suffer from influence of limited range of measurement and sensitivity to light condition and can only generate the sparse 3D map. Besides, the large size and power consumption of these depth sensors render their applications to small robotics, like drones to almost none. Due to the low cost, small size and wide applications of monocular cameras, estimating the dense depth map from a single image has received more attention, and it has been well researched based on deep learning in an endto-end manner recently.

In this paper, we will demonstrate how we can leverage the state-of-the-art AdaBins model to perform monocular depth estimation on indoor spaces by training it on the NYU-Depth-V2 dataset[11].

II. Related Work

a) Physical sensor based solutions: There are two popular sensors-based solutions: RGB-D camera and LIDAR. RGB-D cameras like Kinect, Realsense and Primesense collect depth information by emitting IR(Infrared Light) signal patterns into surroundings and interpret the received signal and in this way RGB-D cameras are capable of capturing pixel-wise depth information. However they have limited detection range (usually from 5.5m to 14.5 meters) and also RGB-D cameras are very sensitive to lighting conditions, especially they can be severely affected by the strong sunlight. LIDAR solutions are able to make long range depth detection but due to their extremely high energy consumption and large size, applications of LIDAR on many medium to small sized platforms are hindered. Meanwhile LIDAR only returns sparse depth map.

b) Geometric constraints based solutions: Representative geometric constraints based solutions are Structure from Motion(SfM) [3] and Stereo Vision Matching [13]. SfM is able to generate a sparse depth map by taking a sequences of images from a single camera and using the correspondences between images and feature matching algorithms. Stereo Vision Matching works similar to how human eyes perceive the world. It requires a calibrated stereo camera setup, and it takes a pair of stereo images as input and by calculating a disparity map from the left and right images and using geometric constraints between images, sparse depth information are calculated. The geometric constraints based solutions have high accuracy once given the exact feature correspondences between images. Their performance strongly rely on the camera setup and the feature matching process, and only produce sparse depth maps.

c) Deep Learning(DL) based solutions: DL based solutions are of three categories by their training methods.

Early representative supervised method by D.Eigen et al. [14] formulates the monocular depth prediction as a regression problem: ground truth depth maps and RGB images are used as to a train CNN based architecture and the disparity between the predicted depth map and the ground truth is used as the supervisory signal. Unsupervised methods are also popular since precise depth maps are hard to acquire. Unsupervised solution proposed by Zhou et al. [15] takes a sequence of images from a single camera as input to a multitask network that learns a pose network and depth network. During training the network uses a pair of consecutive images and use the pose and depth network to reconstruct one image based on the other image, the reconstruction error is the supervisory signal and during prediction, only uses the depth network to infer depth from a single image. Popular semi-supervised solution [16] by R.Garg et al. trains a view synthesis framework using a pair

of images from a stereo camera setup. The multitask network learns to perform an inverse image warping from left camera image to right camera image using the depth network's output and the inconsistency between the warped image and the right camera image is used as the supervisory signal.

Deep learning based solutions have advantages over the traditional methods by faster prediction, denser predicted depth map and easier end-ten-end training pipeline.

III. Dataset

We are using the NYU-Depth-V2[11] dataset. It is comprised of video sequences from a variety of indoor scenes as recorded by both RGB and Depth cameras. Among the different indoor environments, these include basements, bathrooms, bedrooms, kitchens, offices etc. with 464 different indoor scenes in this dataset. Different from the KITTI dataset, another popular dataset for monocular depth training that collects ground truth with LIDAR, the NYU Depth dataset takes monocular video sequences of scenes and the ground truth of depth by an RGB-D camera.

NYU v2 consists of a Labeled Dataset with fine depth details and approx. 2.8 GB in size and the Raw Dataset with more coarse details and roughly 428 GB is size. Given the resource limitation provided to us by Datahub@UCSD, we decided to initially train our model on the Labeled Dataset and then we later retrained it on the Raw Dataset for a single scene type: Office Spaces.

It is worth mentioning that in order to work with this dataset it is required the use of a toolkit[12] to extract the depth maps from the RGB-D camera data.

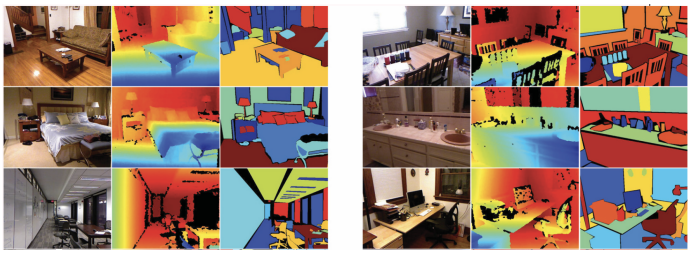


Fig. 1. Sample of NYU-Depth-V2 Dataset

IV. Methodology

In this project we primarily study two state-of-the-arts DL based solutions proposed in recent two years: Vision Tansformer for Dense Prediction (DPT) and Depth Estimation using Adaptive Bins (AdaBins). Both of the two solutions use supervised learning methods to minimize the difference between the predicted depth map and the ground truth depth map. The innovative contributions of DPT and AdaBins models are their use of Vision Transformers at different stages of the networks. Details are described in following paragraphs.

Vision Transformer(ViT) is proposed by Alexey et al. [17] initially for image classification purpose in 2020. Borrowing the idea of the attention mechanism from Natural Language Processing (NLP), ViT works as following: ViT takes a $H \times W \times C$ image as input and divides it into N images patches, each of size $P^2 \cdot C$. (H,W) and (P,P) denote the resolutions of input image and the image patch respectively. Each of the images patches is appended a positional embedding indicating the location of the image patch. Then the positional embedded patches are feed into a transformer encoder consisting of multi-head self-attention (MSA) layers and Multi-layer Perceptron (MLP) blocks; the feature dimensions maintain constant throughout the encoding process. The final outputs of the transformer encoder is a feature map encoding the interactions between image patches through the uses of position embeddings and the MSA layers. The ViT's advantage of global information encoding makes it a great choice for the design of neural architectures that perform the task of monocular depth estimation, since the global information is essential for inferring the relative depth of the objects.

A. Models

a) DPT: DPT model uses ViT for the purpose of dense prediction tasks of monocular depth prediction and image segmentation. In this project we focus on the depth prediction task. DPT model uses a stack of ViT encoders during encoding stage to capture the global information and the interactions between each image patches. The architecture is shown in Figure2: as shown on the left part of Figure2, the DPT model takes a single image as input; the model first divides the image into position embedded image patches (tokens) shown in orange and there is a class token shown in red because the model is borrowed directly from paper [18] which designs the model for classification task. At each stage of the stacked ViT, the output tokens from each ViT encoder layer is reassembled into an image-like feature map of 1/s the spatial resolution of the input image and they are passed into the Fusion blocks which merge the output feature map from previous fusion block and the output from the current reassemble block. The final output of the fusion decoding blocks go into a MLP head for depth prediction task. As shown in the middle, the reassemble block takes the image patches (orange) and a class token (red) from the ViT layer's output; the 'Read' process concatenates the class token onto each of the image patch and the concatenated image patch pass through a MLP to generate the processed patches in yellow; at last the patches are are concatenated together and pass through up-sampling block. As shown on the right, a fusion block takes the output of current reassemble block and passes through a Residual Convolutional Unit and then concatenates it with the output from previous fusion block and finally use another Residual Convolutional Unit and upsample the feature map.

According to the paper of ViT: "Transformers lack some of the inductive biases inherent to CNNs, ..., and therefore do not generalize well when trained on insufficient amounts of data." The training of DPT model requires mixing several datasets to form a mega training dataset. Due to the huge memory requirement, we can not train the DPT model and we only evaluate the model performance with the pretrained weights.

b) AdaBins: The Adabins model uses the techniques of ViT and data quantization: uses a mini ViT to generate a vector of the possible depth bin widths, and uses classification method to express the depth prediction as a linear combination of the depth bin centers. The model consists of two parts: a standard encoder-decoder architecture and the proposed AdaBins Module. As shown in Figure3, the model takes a single RGB image as input and feeds into the standard encoder-decoder block to output an abstract feature map, and then the feature map passes into the AdaBins module. The feature map first passes through a miniViT encoder, a simplified version the ViT and thus does not require as many data during training. The miniViT encoder divides the input feature map into position embedded tokens and feeds through the transformer; the transformer's output is used to produce a vector of possible bin widths (b) through 1 by 1 convolution kernel and a range-depth map (R) through dot product between the processed input feature map and processed output from transformer encoder. Then R goes through a convolutional layer to perform a hybrid regression to get N softmax possibilities for each pixel that correspond to the possible depth bin centers, and at the same time the vector b passes through block (Eq.2) to calculate the bin centers from the N bin widths. And finally the bin centers and the softmax possibilities are combined as a linear combination of the possible N bin center weighted by the N possibilities to get the final depth prediction.

The encoder used in the paper is EfficientNet-b5 for its ability to take high resolution input images and extracts out highly abstract features. We also try several other variants of EfficientNet baseline model as encoder to study the encoder's effects on depth prediction. During training we also uses Adversarial Propagation (AdvProp) [19] to increase the trianing efficienvy. AdvProp increases the model performance by creating adversarial examples (input image with noises added) during training and adding an auxiliary batch normalization to learn the distribution of the noises. In this way the model achieves better performance with the same amount of data.

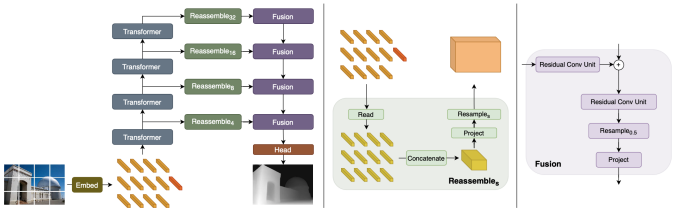


Fig. 2. DPT - Model Architecture

V. Experiments/Results/Discussion

Our first step was to train the baseline model on the reduced NYUV2 Labeled Dataset for 150 epochs. We then wanted to better understand how the model works so we tweaked it at 4 its upsampling decoding layers by adding batch normalization and dropout to them. Finally, we retrained and evaluated five different models as described by the table below and compared their performances.

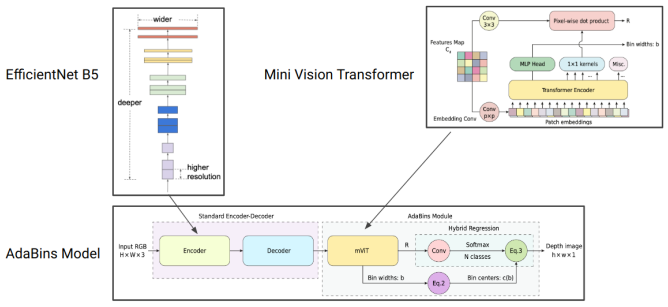


Fig. 3. AdaBins - Model Architecture

	#0 (baseline)	#1	#2	#3	#4	#5
Drop Out Rate	0.0	0.3	0.5	0.0	0.3	0.5
Batch Normalization	True	True	True	False	False	False

Fig. 4. Models trained

This experiment helped us gain more in depth knowledge on the AdaBins architecture and whether model improvements could be made. We reevaluated all six models, compared them to each other and to the fully-trained models, using the weights provided by the authors of the DPT and AdaBins papers [1][2].

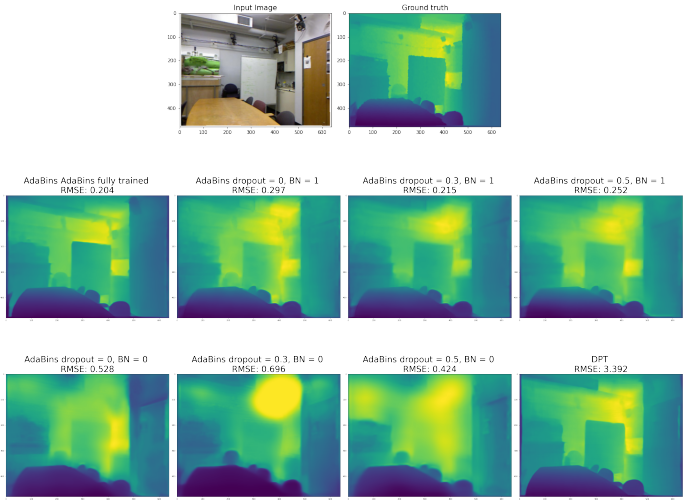


Fig. 5. Models RMSE Performance

In figure 5, we can see how well each model trained measured by the lowest RMSE each achieved and compared it to the fully-trained DPT and Adabins models. It is clear that our models (trained on a smaller dataset) weren't able to replicate the same performance as the DPT and Adabins (fully-trained) models as those were trained on the full raw NYUV2 dataset which is 200x greater than ours. However, what was noticeable was that our model with a 0.3 dropout had a slightly better performance than the baseline model with

no dropout. We believe this small added dropout was able to reduce any minimal overfitting that the model may have been seeing. It would be interesting to see if a model trained on the full dataset also showed such improvement with the same dropout.

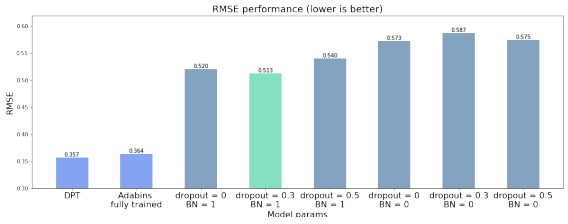
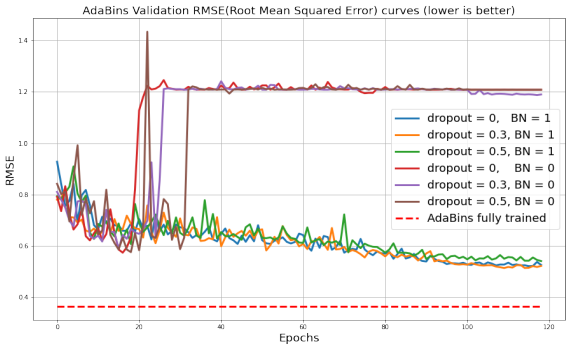


Fig. 6. Model Prediction on Random Image & RMSE

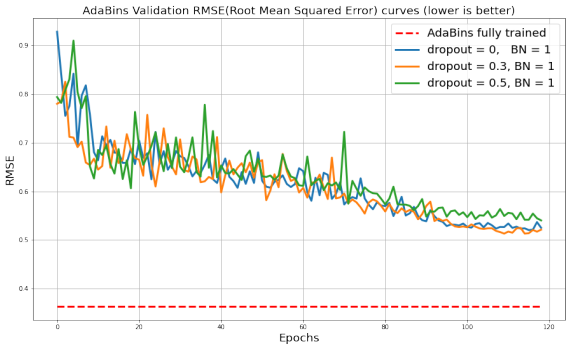
Figure 6 shows the monocular depth estimation of a random image for the different models using the computed weights. Here we compare each depth estimation to the ground truth and calculate their RMSE scores. We notice that the best performing model is clearly the Adabins (fully trained) and that the trained model with a 0.3 dropout comes in second, outperforming the baseline model. This behavior is in tune with the observation we made above where the model with the 0.3 dropout beat the baseline. It is also worth mentioning that by analyzing the DPT depth estimation, the RMSE score doesn't make sense but if one looks at the image itself it seems very similar to the ground truth and the fully-trained Adabins estimation. This is because the DPT model is scaled differently and it's a known issue in their github repository.

Figure 7a shows the RMSE score during training for all models. We immediately noticed that all models without batch normalization exploded around the 30th epoch. This makes sense since batch normalization has the effect of stabilizing the learning process and dramatically reducing the number of training epochs required to train deep networks. It is also there to reduce a models sensitivity to weights. It is possible that all those models generated a set of weights around the 30th epoch that caused the model to explode like that. Figure 7b then shows only the models that had batch normalization and how close each of their performance was and also how far they were from the fully trained model. The only difference between the models here are their dropout layers or lack thereof.

Finally, as part of our presentation review process, we experimented with a few different encoder-decoder architectures. As mentioned, the baseline model uses EfficientNetB5 as its encoder-decoder so we implemented both a more and less complex EfficientNetB6 and EfficientNetB4, respectively. The performance comparison are shown in Figure 8.



(a) All Models



(b) Models w/ Batch Normalization

Fig. 7. Epoch vs. RMSE

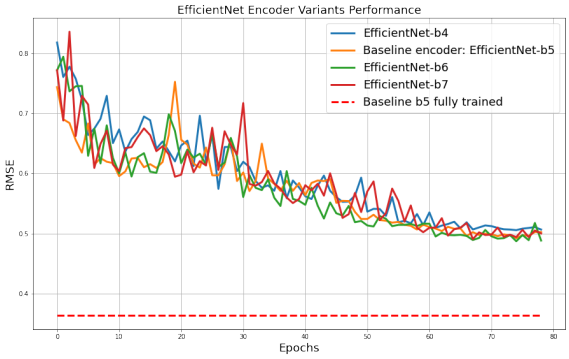


Fig. 8. The RMSE Vs Epochs for diffferent EfficientNet encoders

VI. Conclusion

In this project we investigate two state-of-the-art models on monocular depth estimation task. With the benchmark dataset NYU v2, we trained the AdaBins model and study the effect of dropout and batch normalization on RMSE performance and we also replace AdaBins model's encoder architecture with three variants of EfficientNet baseline to demonstrate the encoder's influence on the model. In the end, there have been a lot of advances in recent years about the architectures of depth estimation models like the ViTs used in the DPT and AdaBins model. In the future we hope more efforts can be denoted to using geometric hints like vanishing points and semantics within an image.

VII. Team Member Contribution

- **Data preparation**: Siyuan
- **Model evaluation**: Siyuan, Linus
- **Accuracy calculation on DPT**: Linus
- **Training script for DPT**: Siyuan, Linus
- **Adabins baseline Implementation**: Siyuan, Linus
- **Adabins model tweaking implementation**: Siyuan, Linus
- **EfficientNet Encoder Variants implementation**: Siyuan
- **Report**: Siyuan, Linus

References

[1] Bhat, S. F., Alhashim, I., and Wonka, P. Adabins: Depth estimation using adaptive bins. arXiv:2011.14141, 2020.

[2] Rene Ranftl, Alexey Bochkovskiy and Vladlen Koltun. Vision Transformers for Dense Prediction. [cs.CV] 24 Mar 2021.

[3] S. Ullman, "The interpretation of structure from motion," Proceedings of the Royal Society of London. Series B. Biological Sciences, vol. 203, no. 1153, pp. 405–426, 1979.

[4] "Cnn-slam, keisuke and tombari, federico and laina, iro and navab, nassir," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 6243–6252.

[5] J. M. Facil, B. Ummenhofer, H. Zhou, L. Montesano, T. Brox, and J. Civera, "Cam-convs: camera-aware multi-scale convolutions for single-view depth," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2019, pp. 11 826–11 835.

[6] R. Wang, S. M. Pizer, and J.-M. Frahm, "Recurrent neural network for (un-) supervised learning of monocular video visual odometry and depth," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 5555–5564.

[7] P. Chakravarty, P. Narayanan, and T. Roussel, "Gen-slam: Generative modeling for monocular simultaneous localization and mapping," in 2019 International Conference on Robotics and Automation (ICRA). IEEE, 2019, pp. 147–153.

[8] C.Godard,O.MacAodha,andG.J.Brostow,"Unsupervised monocular depth estimation with left-right consistency," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 270–279.

[9] Chaoqiang Zhao, Qiyu Sun, Chongzhen Zhang, Yang Tang and Feng Qian. Monocular Depth Estimation Based On Deep Learning: An Overview. [cs.CV] 3 Jul 2020.

[10] OpenAI, GPT3, Retrieved from *https://openai.com/blog/gpt-3-apps/*

[11] OpenAI, Dall-E, Retrieved from *https://openai.com/blog/dall-e/*

[12] NYU Depth Dataset V2, Indoor Segmentation and Support Inference from RGBD Images, Retrieved from *https://cs.nyu.edu/ silberman/datasets/nyu_depth_v2.html*

[13] L. Zou and Y. Li, "A method of stereo vision matching based on opencv," in 2010 International Conference on Audio, Language and Image Processing. IEEE, 2010, pp. 185–190.

[14] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in Advances in neural information processing systems, 2014, pp. 2366–2374.

[15] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1851–1858.

[16] R. Garg, V. K. BG, G. Carneiro, and I. Reid, "Unsupervised cnn for single view depth estimation: Geometry to the rescue," in European Conference on Computer Vision. Springer, 2016, pp. 740–756.

[17] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Syl- vain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929, 2020.

[18] Rene Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. TPAMI, 2020.

[19] Cihang Xie, Mingxing Tan, Boqing Gong, Jiang Wang, Alan Yuille, Quoc V. Le Adversarial Examples Improve Image Recognition, CVPR 2020

Replies To Critical Reviews

A. Critical review from Group 26

It seems that the final prediction is a linear combination of several outputs. So what's the range of this model can predict? Is this determined by the environment the model works in?

Our response:

Great question! Yes the range for NYU v2 dataset is (0.001, 15) meters, and this range is determined by the dataset, for outdoor driving dataset like Kitti the range will be larger.

B. Critical review from Group 26

The dataset can be memory intensive. How to achieve a balance between model accuracy and the size of the dataset?

Out response:

To achieve best performance given the limited disk memory on datahub, we tried to use a more concentrated dataset to train the model; Instead of using all the training data of many scenes, we extracts out all the training images for office scene and also include the corresponding data from the NYU Raw dataset.

C. Critical review from Group 23

I think it would be interesting to propose a modification to the model architecture, as opposed to just adding a dropout layer and comparing results. Maybe replace auto-encoder with a state of the art autoencoder architecture to compare results.

Out response:

We think this is a good idea worth trying, so besides the EfficientNet-b5 encoder used in the original model, we also replace it with downscaled encoder EfficientNet-b4, up-scaled encoder EfficientNet-b6 and further up-scaled encoder EfficientNet-b7 and their performance are shown in Figure 8.