A NOVEL LIGHTWEIGHT NETWORK FOR FAST MONOCULAR DEPTH ESTIMATION

*Tim Heydrich[1], Yimin Yang[1], Xiangyu Ma[2], Yu Liu[2], Shan Du[3]**

[1]Lakehead University,
Department of Computer Science,
Thunder Bay, ON, Canada

[2]Tianjin University,
School of Microelectronics,
Tianjin, China

[3]The University of British Columbia Okanagan,
Department of Computer Science, Mathematics, Physics and Statistics,
Kelowna, BC, Canada

ABSTRACT

Depth estimation is of growing interest in many sectors, from robotics to wearable augmented reality gears. Monocular depth estimation attracts more attention due to its cost efficiency and low complexity. Most recent research has developed very large and resource intensive networks which are not suitable for small systems with limited resources. In this paper, we propose a lightweight network which leverages the advantages of dimension-wise convolutions and depthwise separable convolutions to reduce complexity in the architecture. In particular, the proposed depth estimation architecture utilizes a novel DICE unit-based encoder, optimized for a lightweight encoder-decoder structure. Furthermore, we propose a DICE unit-based decoder structure as well as an optimized depthwise separable convolution-based decoder. Both decoders follow a similar five-layer architecture. In the experiments, we have demonstrated the effectiveness of the proposed architecture as well as the comparison between the two proposed decoders. Our novel lightweight network has a significant decrease in both size and complexity at a marginal cost to accuracy when compared to other state-of-the-art lightweight networks.

Index Terms— computer vision, depth estimation, deep learning, lightweight

1. INTRODUCTION

Depth estimation is a crucial task for many areas from scene reconstruction [1], to augmented reality (AR) and robotics. Due to the cost efficiency and the ease of use when compared to more complex systems such as LiDAR or stereo depth, monocular depth estimation is preferred for lots of small robotic platforms as well as wearable AR gears.

Most research in recent years has largely focused on heavy deep learning models to increase the accuracy and quality of the depth estimation [2], [3], [4]. These large and complex neural networks produced good results at the expense of computational resources and time. The networks need large amounts of RAM to be stored but also have a very slow inference time which makes the use in critical systems questionable. This shows the need for a lightweight approach. The most notable research in this area is FastDepth [5], where the network architecture relies on depthwise separable convolutions to reduce the network complexity. The auto-encoder structure is widely adopted by most monocular depth estimation networks. FastDepth utilizes the MobileNet [6] architecture as an encoder and their own depthwise separable convolution based decoder to produce the auto-encoder structure. With our proposed method we aim to further reduce the size and computational complexity to make monocular depth estimation more accessible to resource weak systems. In this paper, we are able to further reduce the size and complexity by using the DICE unit structure [7]. With the use of the DICE unit, we are able to create a novel architecture that reduces FastDepth's size by two thirds and the computational complexity by half with only a minor reduction in accuracy. In particular, we propose a DICE unit-based encoder and two custom designed decoders. The first decoder utilizes DICE units and the second one utilizes depthwise separable convolutions. As mentioned above, this proposed architecture is capable of greatly reducing both size and complexity of current state-of-the-art lightweight networks. Our architecture is able to reduce the network size by almost 70% and the complexity by 50% while

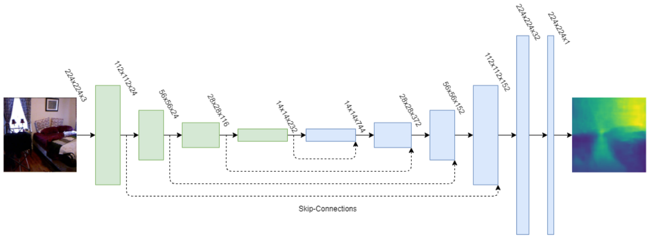

Fig. 1. Overall architecture overview. Here the encoder is in green and the decoder in blue. Both proposed decoders utilize the same structural idea.

*Corresponding author: Shan Du (shan.du@ubc.ca). This work was supported by the University of British Columbia Okanagan [GR017752], Lakehead University [11-50-16112406], and Vector Institute.

still achieving competitive accuracy with a reduction of less than 6% when compared to other state-of-the-art lightweight networks.

Our contributions are summarized as follows: 1) A DICE unit-based architecture is proposed to optimize the performance and size of the encoder. 2) A new decoder is proposed which is composed of DICE units and is capable of achieving competitive results while creating a fully DICE-based network. 3) A second decoder architecture is proposed based on depthwise separable convolutions. When compared with FastDepth decoder, it provides modified layer dimensions and skip-connections to achieve optimal information transfer from the encoder.

2. RELATED WORK

2.1. Depthwise Separable Convolution

The FastDepth architecture [5] relies heavily on depthwise separable convolutions which helped them reduce the size and complexity. However, depthwise separable convolutions have some drawbacks when used in large scale network structures. Depthwise separable convolutions initially proposed in [8] have the ability to greatly reduce both the size and complexity of regular convolutions at a slight expense to accuracy. They are able to do so by using rules of matrix multiplication to split regular convolutions in two, one depthwise and one pointwise convolution. The depthwise convolution performs the dimensional reduction while the pointwise modifies the channel count and gives the channel wise relations. This approach is very effective, however, the use of pointwise convolution to yield channel wise relations creates a computational bottleneck as stated by [7]. This bottleneck shows the need and opportunity to further reduce the complexity of depthwise separable convolutions. DICE units [7] give a solution that is capable of reducing the complexity at no significant cost to accuracy.

2.2. DICENet and DICE Units

In order to reduce the computational bottleneck, Mehta et al. [7] proposed the DICE unit and utilized said unit to form the DICENet architecture. DICE units cannot completely remove the need for pointwise convolutions as they are inherently needed to create the wanted channel dimension. However, they are able to reduce the amount of pointwise convolutions needed as well as create a structure to give channel wise relations. Each DICE unit consists of a dimension-wise convolution and a dimension-wise fusion. The dimension-wise convolution computes the dimension-wise reduction and relation by preforming depth-wise, width-wise and height-wise convolutions on independent branches whose output is then concatenated along the depth dimension. Dimension-wise fusion then utilizes that concatenation to produce the channel relation.

2.3. Network size and complexity

When designing lightweight networks, both the size and the computational complexity are of great importance. The network size describes how much storage and random-access memory (RAM) a neural network needs to operate. One of the main attributes that can be used to gauge the size of a neural network is the number of parameters. It needs to be noted that there are other factors that determine the RAM requirements of a neural network such as number of activations as well as what precision is used to store each of the values. One of the goals when creating a lightweight neural network is a low number of parameters to reduce the size. Additionally, the number of parameters is also related to the complexity of a neural network. Most often this complexity is measured in terms of multiply-accumulate (MAC) operations [9]. These MAC operations describe the computational complexity of a neural network as it determines how many operations are performed during each forward pass of a neural network. Reducing the number of MACs is the main goal when designing lightweight networks as it results in faster performance on systems with low computational resources. A lower MAC count also results in lower power consumption, this is key for systems where power consumption is a major concern.

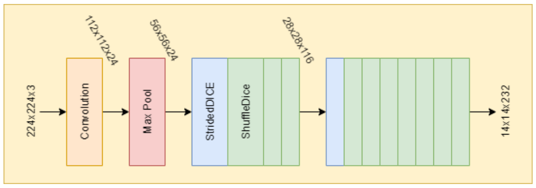

Fig. 2. Optimized 4-level encoder architecture overview. Color coding in the diagram is used to avoid repetition of layer labels.

3. PROPOSED NETWORK ARCHITECTURE

Our proposed fully convolutional encoder-decoder structure relies on the encoder to extract features which are then upsampled and combined by the decoder to produce the final depth estimation. An overview of the architecture can be seen in Figure 1. The figure highlights the flow through the network as well as the skip-connections. Furthermore, the output dimensions of each layer are displayed as well. As further described in Section 3.1, the second layer that is depicted in the diagram is a max-pooling layer, which was included in both Figure 1 and Figure 2 to highlight the size reduction in that step as well as for the additional skip-connections it provides.

3.1. Encoder

The encoder used in our network is originated from the architecture proposed in [7] but specifically designed for depth estimation. This choice was made due to the fact that DICENet can compete with MobileNet but offers reduced size and computational complexity in many applications such as object detection and segmentation. DICENet was originally designed for image classification, in order to preserve the general DICENet structure, we utilized DICEBlocks that have the same internal structure as the ones used by DICENet. We propose the use of a four-block structure. These blocks consist of various amounts of DICE units each, as is lined out in [7]. Our proposed structure, Figure 2, is capable to take advantage of the pre-trained weights that exist for DICENet while being more lightweight and streamlined for feature extraction instead of image classification. The choice to utilize four blocks comes from the fact that utilizing more blocks lead to similar overall accuracy for depth estimation. In order to reduce size and complexity, we propose to use the minimum viable number of blocks. Our novel encoder architecture has a significant size and complexity reduction when compared to the encoder of other state-of-the-art lightweight methods such as FastDepth. Furthermore, each of the encoder blocks also provides a good feature output for skip-connections to the decoder. Skip-connections are commonly used for depth-estimation as they provide feature transfer between the encoder and the decoder. We utilize the output of each encoder block as the features passed to the decoder.

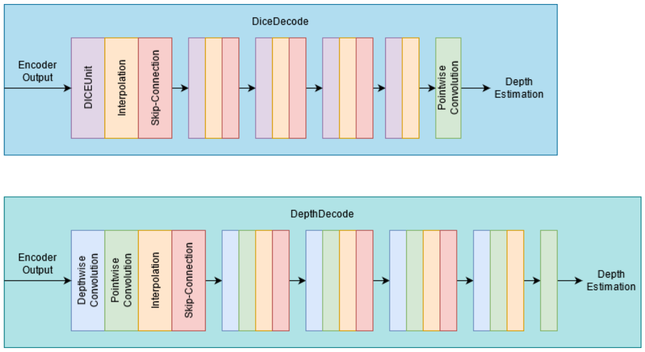

Fig. 3. Decoder architecture overview. The DiceDecode utilizes DICE units whereas the DepthDecode utilizes depthwise separable convolutions. Color coding in the diagram is used to avoid repetition of layer labels.

3.2. Decoder

The decoder's objective is to gradually up-sample and combine the feature maps together to form a dense depth map as output. We developed and tested two different decoders for this paper named DepthDecode and DiceDecode. The first decoder we propose, DiceDecode, consists of a series of DICE units, interpolation and concatenative skip-connections. The decoder is grouped into five similarly structured blocks, the first four blocks consist of a DICE unit followed by a nearest neighbor interpolation and a skip-connection. The fifth block does not utilize skip-connections.

The output layer of the decoder is a single pointwise convolution which transforms the channel dimension into the single output dimension of the depth estimation. DiceDecode is capable of producing adequate results while preserving the overall low complexity of the network. After several experiments and network analysis, we believe however that the DICE units used in our decoder can be replaced with depthwise separable convolutions which could further reduce the complexity and maintain the accuracy. We came to this conclusion since in the original work [7], the DICE units were commonly paired with regular convolutions. In the DICENet proposed by [7], as well as our encoder, the DICE units are combined with regular convolutions into block-like structures, such as the Strided-DICE. This leads to the assumption that DICE units by themselves can be outperformed by depthwise separable convolutions. Since the utilization of regular convolutions is not feasible in a small architecture, like our decoder, without drastically increasing complexity, we decided to propose a second decoder, DepthDecode. This second decoder we propose follows a similar overall architecture but the DICE units are replaced by depthwise separable convolutions. DepthDecode still follows the five-block architecture where the first four blocks consist of a depthwise convolution followed by a pointwise convolution, an interpolation and a skip-connection. Similar to DiceDecode, the fifth layer does not have a skip-connection. The output layer of our DepthDecode is also a single pointwise convolution. It needs to be noted that our novel DepthDecode differs from FastDepth decoder in both layer dimension and skip-connection utilization. A detailed architecture overview of both proposed decoders can be seen in Figure 3. As mentioned above, in order to further optimize network performance skip-connections are used. These skip-connections are used to pass information from the four levels of the encoder to the decoder. As mentioned, the decoders consist of five modules, this results in the last module not receiving any information from the encoder. The decision to maintain a five-layer decoder was made to maintain a 1:1 relation for the spatial dimensions of the input and the output. If the decoder were to only consist of four layers, the output would be half the size of the input image which is not desirable in most cases. The skip-connections employed use concatenation as it has been proven in various previous works [5], [3] that they achieve better results when compared to additive skip-connections.

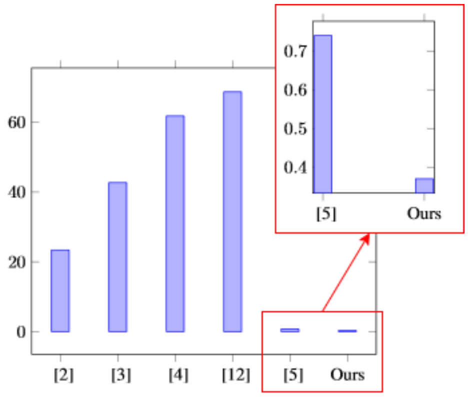

Fig. 4. Complexity (MACs) comparison between our proposed network and current state-of-the-art networks. The model chosen to represent our proposed network is the one utilizing the DepthDecode decoder.

3.3. Loss Function

After extensive testing, we utilize the L_1 loss function for our final model due to its simplicity. The training for [5] was performed similarly to [10], both used L_1 as well. In [10] their experiments showed that it produced better results when compared with the L_2 loss, MSE, and the Reversed Huber loss. It is formulated as:

$$L_1 = \sum_{i=1}^{n} |y_{true} - y_{pred}| \tag{1}$$

where n is the number of samples, y_{true} denotes the ground truth target, and y_{pred} is the value predicted by the network.

4. EXPERIMENTS

4.1. Experimental Setup

The network is trained and evaluated on the NYU Depth v2 dataset [11] using the official train/test split. The training set up is equivalent to the one used by [5]. This means the use of SVG as optimizer with a learning rate of 0.01, a momentum of 0.9 and a weight decay of 0.0001. The encoder utilizes pre-trained image classification weights. The training and evaluation of this model is performed on a Nvidia GTX 1080Ti.

4.2. Pruning

Pruning is a very common technique used to reduce the size and complexity of neural networks. Some authors utilized pruning as part of their proposed model to achieve lower parameter and MAC count. We did not apply any pruning to our model as we believe that pruning can be applied to most networks to reduce complexity and is not an inherent property of any specific network architecture. Due to this, we will compare our network to unpruned networks or the unpruned version of networks which included pruning as part of their architecture proposal.

4.3. Comparison to State-of-the-Art

The comparison to state-of-the-art compares our proposed architecture, encoder and both decoders, not only to state-of-the-art heavy networks such as [2], [3], [4] but also to the current state-of-the-art lightweight network [5]. The models are evaluated based on their Absolute Relative Error (Abs Rel), Root Mean Squared Error (RMSE), Absolute Relative Distance (Abs Rel), Mean Squared Logarithmic Error (Log_{10}) as well as the $\delta 1$ metric. A visual comparison of the network complexity in MACs is given in Figure 4.

Table 1. Comparison with current state-of-the-art heavy networks. For $\delta 1$ higher is better and for every other criteria lower is better.

On NYU Depth V2	MACs (G)	Abs Rel	Log_{10}	RMSE	δ 1
Eigen et al. [2]	23.4	0.158	-	0.641	0.769
Xian et al. [4]	61.8	0.155	0.066	0.660	0.781
Liana et al. [3]	42.7	0.127	0.055	0.573	0.811
DORN (Fu et al 2018) [12]	68.17	**0.115**	**0.051**	**0.509**	**0.828**
Wofk et al. [5]	0.74	-	-	0.599	0.775
Ours (DiceDecode)	0.45	0.181	0.079	0.663	0.704
Ours (DepthDecode)	**0.37**	0.188	0.077	0.654	0.718

The detailed comparison with all relevant state-of-the-art networks can be seen in Table 1. A quick comparison with our immediate competitor FastDepth can be seen in Table 2. From these comparison, the achievements are quite obvious. Our novel architecture is able to further reduce the size achieved by FastDepth without a significant loss to accuracy. From the comparison, it also becomes apparent that a pure use of DICE units in the decoder is sub-optimal when compared to depthwise separable convolutions. As we suspected during our experiments, DICE units can outperform depthwise separable convolutions when used along side regular convolutions, however when used independently they are inferior.

Table 2. Quick in depth comparison of our proposed architectures with the current state-of-the-art lightweight network FastDepth.

On NYU Depth V2	MACs (G)	Parameters (M)	RMSE	$\delta 1$
Wofk et al. [5]	0.74	3.93	**0.599**	**0.775**
Ours (DiceDecode)	0.45	1.56	0.663	0.704
Ours (DepthDecode)	**0.37**	**1.21**	0.654	0.718

A visual comparison between our proposed work and our immediate competitor FastDepth is given in Figure 5. From the comparison, we can see that there is some minor loss in detail. Overall, however, our network can still provide adequate depth estimation at a significant size and complexity reduction. We believe that the size reduction achieved here is worth the slight loss in accuracy, less than 6%, as it is able to reduce the model size by almost 70% with respect to the size of FastDepth and the complexity by 50%.

4.4. Ablation Study: Loss Function

As mentioned in 3.3, we performed extensive testing on various loss functions. We were able to eliminate certain ones due to the work provided in [10]. They already established that the L_1 loss is able to produce better results than the L_2 loss, MSE and the Reversed Huber. We wanted to further explore different loss functions. In an attempt to more exhaustively determine the effectiveness of the L_1 loss, we explored the following loss functions: L_2, L_1 + SSIM, L_1 + L_2, L_2 + SSIM, where SSIM refers to the Structural Similarity. The results of our experiments can be seen in Figure 6. From the figure, it is apparent that the L_1 loss is the best choice as it is not outperformed by any of the other loss functions. There were some that performed on par with the L_1 loss, however, since our goal was to create a lightweight approach we chose the simplest and most effective loss function, L_1 loss.

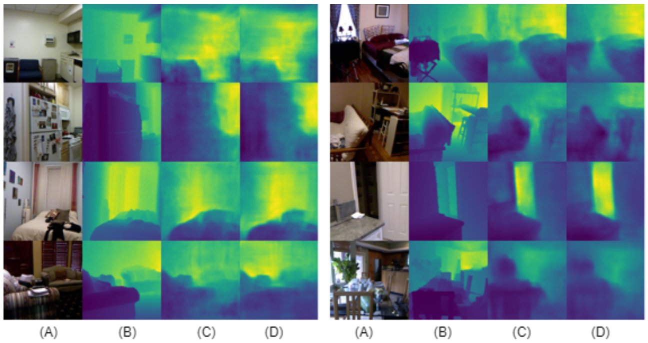

Fig. 5. Visual comparison between our proposed work and FastDepth. (A) is the RGB input image to the network, (B) is the ground truth target, (C) is FastDepth's result and (D) is our result.

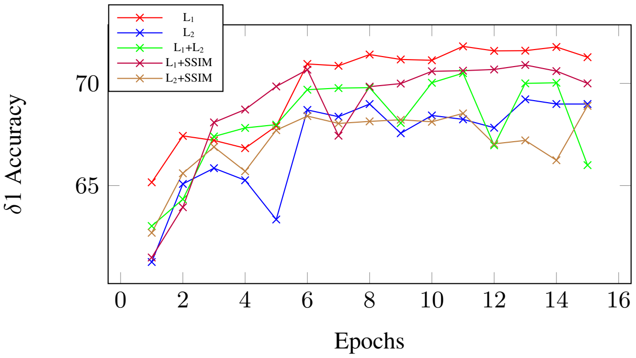

Fig. 6. Loss function comparison per epoch based on the $\delta 1$ testing accuracy. Results from training our proposed encoder with our DepthDecode decoder.

5. CONCLUSION

In this work, we proposed a novel lightweight architecture for monocular depth estimation. We propose a novel encoder architecture as well as two different decoder architectures. Our model is able to greatly reduce the size and complexity of current state-of-the-art lightweight networks. There is a slight reduction in accuracy but we believe that the trade off is worth it. Furthermore, we believe that this model would profit from unsupervised learning environments and is more suitable for real-time learning. Despite monocular depth estimation being the main focus of the network proposed here, we believe that it can be of significant use in other branches as well.

6. REFERENCES

[1] J. Lehtinen, J. Munkberg, J. Hasselgren, S. Laine, T. Karras, M. Aittala, and T. Aila, “Noise2noise: Learning image restoration without clean data,” *Proceedings of the 35th International Conference on Machine Learning (ICML)*, vol. 80, pp. 2965–2974, 2018. [Online]. Available: https://arxiv.org/abs/1803.04189

[2] D. Eigen and R. Fergus, “Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture,” *Proceeding of IEEE International Conference on Computer Vision (ICCV)*, 2015. [Online]. Available: http://arxiv.org/abs/1411.4734

[3] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, “Deeper depth prediction with fully convolutional residual networks,” *Proceedings of Fourth International Conference on 3D Vision (3DV)*, 2016. [Online]. Available: http://arxiv.org/abs/1606.00373

[4] Z. Hao, Y. Li, S. You, and F. Lu, “Detail preserving depth estimation from a single image using attention guided networks,” *Proceedings of International Conference on 3D Vision (3DV)*, 2018. [Online]. Available: http://arxiv.org/abs/1809.00646

[5] D. Wofk, F. Ma, T. Yang, S. Karaman, and V. Sze, “Fastdepth: Fast monocular depth estimation on embedded systems,” *Proceedings of International Conference on Robotics and Automation (ICRA)*, pp. 6101–6108, 2019. [Online]. Available: https://arxiv.org/abs/1903.03273

[6] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “Mobilenets: Efficient convolutional neural networks for mobile vision applications,” *arXiv*, 2017. [Online]. Available: http://arxiv.org/abs/1704.04861

[7] S. Mehta, H. Hajishirzi, and M. Rastegari, “Dicenet: Dimension-wise convolutions for efficient networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2020. [Online]. Available: https://arxiv.org/abs/1906.03516

[8] L. Sifre, “Rigid-motion scattering for image classification,” Ph.D. dissertation, Ecole Polytechnique, 2014.

[9] J. Chang, Y. Choi, T. Lee, and J. Cho, “Reducing mac operation in convolutional neural network with sign prediction,” *Proceedings of International Conference on Information and Communication Technology Convergence (ICTC)*, pp. 177–182, 2018.

[10] F. Ma and S. Karaman, “Sparse-to-dense: Depth prediction from sparse depth samples and a single image,” *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 2018. [Online]. Available: http://arxiv.org/abs/1709.07492

[11] P. K. Nathan Silberman, Derek Hoiem and R. Fergus, “Indoor segmentation and support inference from rgbd images,” *Proceedings of European Conference on Computer Vision (ECCV)*, pp. 2002–2011, 2012.

[12] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao, “Deep ordinal regression network for monocular depth estimation,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. [Online]. Available: http://arxiv.org/abs/1806.02446