VIETNAM NATIONAL UNIVERSITY – HOCHIMINH CITY
THE INTERNATIONAL UNIVERSITY
SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

INTERNSHIP REPORT

by

NGUYỄN TIẾN CƯỜNG

AN INVESTIGATION ON A CNN-BASED POLYP SEGMENTATION ALGORITHM FOR MEDICAL IMAGES

School of Computer Science and Engineering
International University, VNU-HCM
September, 2021

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

I. TASK DESCRIPTION

In this internship, I was assigned to implement some state-of-the-art methods in medical image processing. Particularly, my task during the internship was to implement some methods on the polyp segmentation task. The importance of solving this task is emphasized by the fact that polyps are the most important cause of the colorectal cancer. Thus, segmentation and detection of these polyps in early stages of caner will increase the chance of survival. In this task, given a colonoscopy image, the computer algorithm should segment out the region containing the polyp to notify the human experts, thereby reducing human errors in the detection of such polyps (see Figure 1).

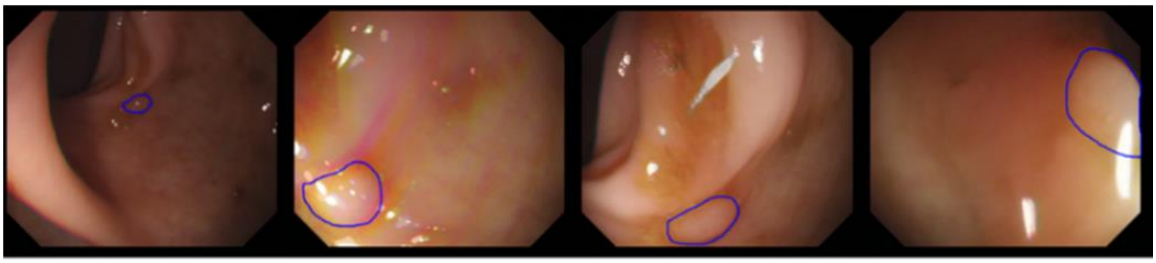

Figure 1 - A sample of typical polyps.

In order to tackle this automatized medical process, I perform an extensive study of the current best methods, the techniques required to improve them, and the frameworks to implement them as well as trying to balance the speed and accuracy trade-offs if possible. After an extensive study, I have chosen to implement the following methods: HarDMSEG [1], UACANet-S and UACANet-L [2], CANet-S and CANet-L [2], and PraNet [3]. The reported experimental results of these methods is very competitive. However, to verify the results and giving an objective comparison among these methods, I re-implemented and refactorized them into one application with PyTorch, a deep learning framework, and

PyYAML, a library which help the developer get over the tedious process of configuring the parameters of the application, to perform training and evaluation on these methods' speed and accuracy. In addition, the datasets of polyp images with their respective ground-truths are quite rare. Therefore, I learnt to implement some data augmentation techniques to make the deep learning models more generalizable to the evaluation data.

II. Method Description (Technical report alert!!!!)

1.1. Deep learning for medical image segmentation

For medical image segmentation tasks, supervised learning is the most popular method since these tasks usually require high accuracy. Image semantic segmentation aims to achieve pixel classification of an image. For this goal, researchers proposed the encoder-decoder structure with CNNs, one of the most popular end-to-end architectures, such as fully convolution network (FCN) [15], U-Net [16], Deeplab [17], etc. In these structures, an encoder is often used to extract image features while a decoder is often used to restore extracted features to the original image size and output the final segmentation results. Although the end-to-end structure is pragmatic for medical image segmentation, it reduces the interpretability of models.

The first high-impact encoder-decoder structure, the U-Net proposed by Ronneberger et al. [16] has been widely used for medical image segmentation. The main idea is to use the feature map from lower encoder layers to refine the output of deeper decoder layers. This one architecture has greatly impacted the way researchers design many state-of-the-art methods in medical image segmentation. Many of the recent proposed methods [16] [2] [3] [1] on polyp segmentation task, which is the focus of my internship, also follows the basic idea of designing a U-shaped architecture. Figure 5 shows the U-Net architecture.

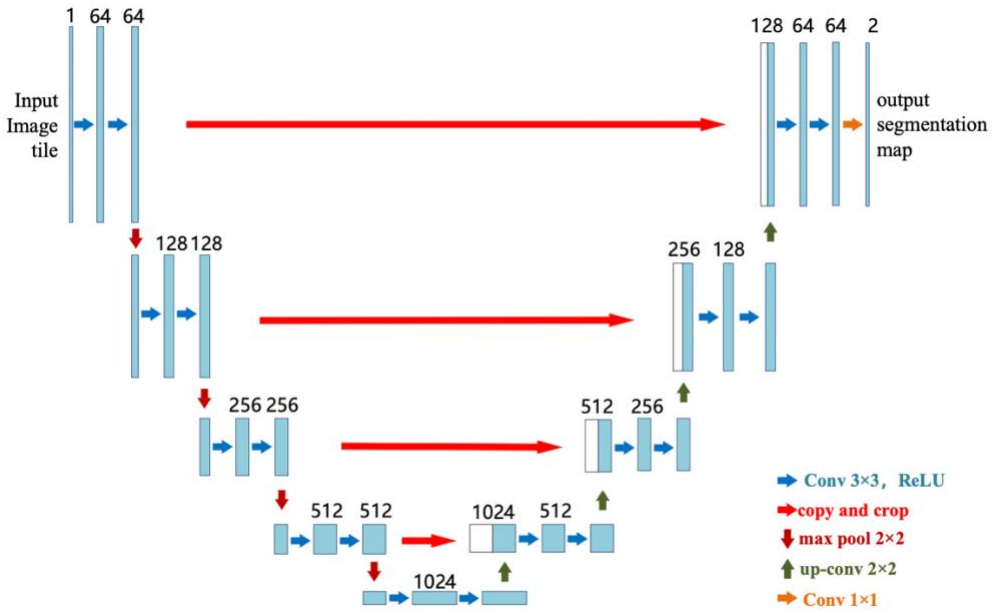

Figure 2 – U-Net architecture.

2. Polyp Segmentation Datasets and Augmenting Techniques

A part of my internship work is to collect the datasets for training and testing model performance. The datasets that I collected and used for this internship include: Kvasir-SEG [18], CVC-ClinicDB [19], CVC-ColonDB [20], CVC-300 [21], ETIS-LaribPolypDB [21].

However, the total number of training images of these datasets is only 1451 images. Learning to generalize under such few training labels is a hardly possible task for any machine learning models. Thus, I have implemented some data augmentation techniques to enrich the training data to enable sufficient generalization on the testing set of the aforementioned datasets. The augmentation techniques [22] that I used for model training include:

- *Random crop*: In this augmentation technique, a region of the input image, with probability *p*, is cropped and rescaled to original image's resolution. This technique aims at rescaling the object of interest of one training image

to make the model robust to different target object scales since we are not guaranteed to observe the same object with the same size in real-world data, e.g., the testing set may contain the same training image but with the target object zoomed in or out.

- *Random flipping*: The input image will be flipped horizontally, vertically, or both during training to produce different perspective of the same training image.
- *Random rotate*: Rotation augmentations are done by rotating the image right or left on an axis between 0° and 359°. This, again, helps the model to become more robust against different capturing angles of one image.
- *Random brightness, contrast, and sharpness*: The brightness, contrast, and sharpness of the training image is randomly changed to make the model focus on the embedded semantics of the target object of interest in the training image.
- *Random dilation and erosion*: The dilation and erosion image processing algorithms are randomly applied with pre-specified kernel sizes to produce different training images.

3. Evaluation of Models

In this internship, part of my work was to implement various state-of-the-art models on medical image segmentation and perform evaluation to find a method which has potential for further improvements. In this report, I conducted various experiments with 6 models, *i.e.*, HarDMSEG [1], UACANet-S and UACANet-L [2], CANet-S and CANet-L

[2], and PraNet [3], for comparison. All results in this report are retrieved from the experiments conducted on an Intel Core i9 9-th generation machine with one NVIDIA RTX Quadro 6000 GPU if not explicitly mention. The 6 methods for method evaluation were trained for 240 epochs using Adam optimizer with 0.0001 learning rate. The augmentation methods applied to all of these methods are the same, as reported in Section IV.2.

In addition to training, I was also finding an effective metrics to evaluate the segmentation result of these methods. The two most widely used metrics are mean dice coefficient (mDice) and mean intersection-over-union (mIoU). The chosen metric for my experiments was mDice. The reason for this choice rather than mIoU is that this metric is better than mIoU at measuring average performance while mIoU penalize negative instances more severe [23]. Intuitively, we can compare the difference between mIoU and mDice as the difference between L2 loss and L1 loss, with L2 loss tends to penalize the largest error more than L1. The formula for dice coefficient is given by the following equation

$$Dice = \frac{2TP}{2TP + FP + FN}$$

where TP, FP, and FN represents true positive pixels, false positive pixels, and false negative pixels, respectively. The result of the experiment is shown in Table 2.

Table 2 below is the evaluation result of the previously mentioned models on 5 different datasets.

Model	CVC-300	CVC-ClinicDB	Kvasir-SEG	CVC-ColonDB	ETIS-LaribPolypDB
CANet-S	<u>0.888</u>	<u>0.907</u>	<u>0.901</u>	**0.774**	0.697
CANet-L	0.905	0.924	0.905	0.767	0.682
UACANet-S	0.901	0.914	0.904	0.765	0.701
UACANet-L	**0.913**	**0.930**	0.903	<u>0.759</u>	<u>0.646</u>
PraNet	0.899	0.911	0.902	0.769	0.703
HarDMSEG	0.896	0.908	**0.909**	**0.774**	**0.759**

Table 1 - Comparison of 6 mentioned methods on 5 datasets using mDice. The most effective on each dataset is bold whole the least effective is underlined.

From Table 2, we can see that, for different datasets, the mDice differences between the most and the least effective is negligible, *i.e.*, within 0.030 margin, with an exception on the ETIS-LaribPolypDB dataset where the difference is 0.113. This shows that the performance differences amongst different methods are very negligible. Thus, to choose a method for improvements, I decided to measure the speed of these methods on images of size 352x352 to pick out the one with highest frames-per-second (FPS). The speed evaluation is shown in Table 3.

Model	Running time in FPS
CANet-S	215.4721
CANet-L	87.3578
UACANet-S	218.7881
UACANet-L	86.7280
PraNet	212.7923
HarDMSEG	353.8707

Table 2 - Efficiency in FPS of 6 models on images of size 352x352.

From Table 3, it is undeniable that HarDMSEG is, indeed, the fastest method with competitive accuracy with the best model with differences of 0.017 on CVC-300 and 0.022 on CVC-ClinicDB (refer to Table 2) while gaining top position on Kvasir-SEG, CVC-ColonDB, and ETIS-LaribPolypDB. Therefore, I choose HarDMSEG as a potential method for improvements.

4. Improving a model: HarDMSEG

4.1. Replacement of the baseline network of HarDMSEG

Although HarDMSEG is very effective in segmenting object as well as efficient with its fast inference with a speed of 353.8707 FPS on an Intel Core i9 9-th gen machine with NVIDIA Quadro RTX 6000 GPU, it is still more desirable to improve its speed with a small trade-off in accuracy to facilitate larger scale deployment of the model. Thus, in this internship, I replaced the HarDNet68 baseline [13] of HarDMSEG as reported in [1]

with the depth-wise separable baselines of HarDNet. The Table 4 and Table 5 below shows the mean dice result and the inference speed, respectively, of the HarDMSEG model with HarDNet39ds and HarDNet68ds as baselines in comparison to the original version of HarDMSEG with HarDNet68 baseline.

Baseline of HarDMSEG	CVC-300	CVC-ClinicDB	Kvasir-SEG	CVC-ColonDB	ETIS-LaribPolypDB
HarDNet39ds	0.859	0.896	0.891	0.723	0.641
HarDNet68ds	0.853	0.894	0.891	0.735	0.693
HarDNet68	**0.896**	**0.908**	**0.909**	**0.774**	**0.759**

Table 3 – mDice measure of the segmentation result of HarDMSEG with different baselines.

Baseline of HarDMSEG	Running time in FPS
HarDNet39ds	677.7053
HarDNet68ds	483.5103
HarDNet68	353.8707

Table 4 - HarDMSEG inference speed with different baselines on 352x352 images.

Table 5 clearly show that the depth-wise separable version of HarDNet significantly improve the efficiency of HarDMSEG. However, as shown in Table 4, the segmentation map outputted by the model deteriorates, with a huge margin of 0.118 and 0.066 in mDice between the HarDNet68 baseline and the HarDNet39ds and HarDNet68ds baselines on the ETIS-LaribPolypDB dataset, respectively. Although suffering from a huge accuracy decrease, the HarDMSEG model with HarDNet39ds and HarDNet68ds baselines somewhat preserves the effectiveness of the original HarDMSEG model with HarDNet68 baseline on CVC-ClinicDB, Kvasir-SEG, and CVC-ColonDB. This, in turn, suggests that

these more light-weighted depth-wise separable versions of HarDMSEG could be improved to approach the effectiveness of the original HarDMSEG model with small trade-off in inference time. The improvement made in this internship will be presented in the next section.

4.2. Adding the attention module

Recently, the deep learning research community has seen an introduction of a mechanism called ‘attention’ for deep neural networks method in natural language processing (NLP) tasks [24]. This ‘attention’ idea is inspired from the biological fact that when we, as human, read a long sentence, we only focus on (*i.e.*, are attentive to) a local region of the text to infer its overall meaning. Similarly, the attention mechanism in deep learning, intuitively, imitates this activity of the human brain for strengthening the use of deep neural networks in NLP problems. This attention mechanism was proven to be effective for NLP problems [24].

Seeing the undeniable effectiveness of the attention mechanism, the deep learning research community in computer vision has been shifting from fully convolutional neural networks to a mix between the attention mechanism and convolutional operators. The shift is quite reasonable because the human brain is also attentive to only a local region of the image instead of scattering its attention. Some fully convolutional neural networks models, after applying attention mechanism, have seen an improvement over the original one without attention [25].

Inspired by the effectiveness of the attention mechanism, I used it to further strengthen the depth-wise separable versions of HarDMSEG. The attention module I used

is CBAM [25]. This CBAM module is inserted after each HarDBlock of the HarDNet baseline for enabling to model to focus on a local region of the input image (see Figure 6 for illustration). In the polyp segmentation task, the goal of adding attention module was to direct the model's attention to regions with polyps in the image.

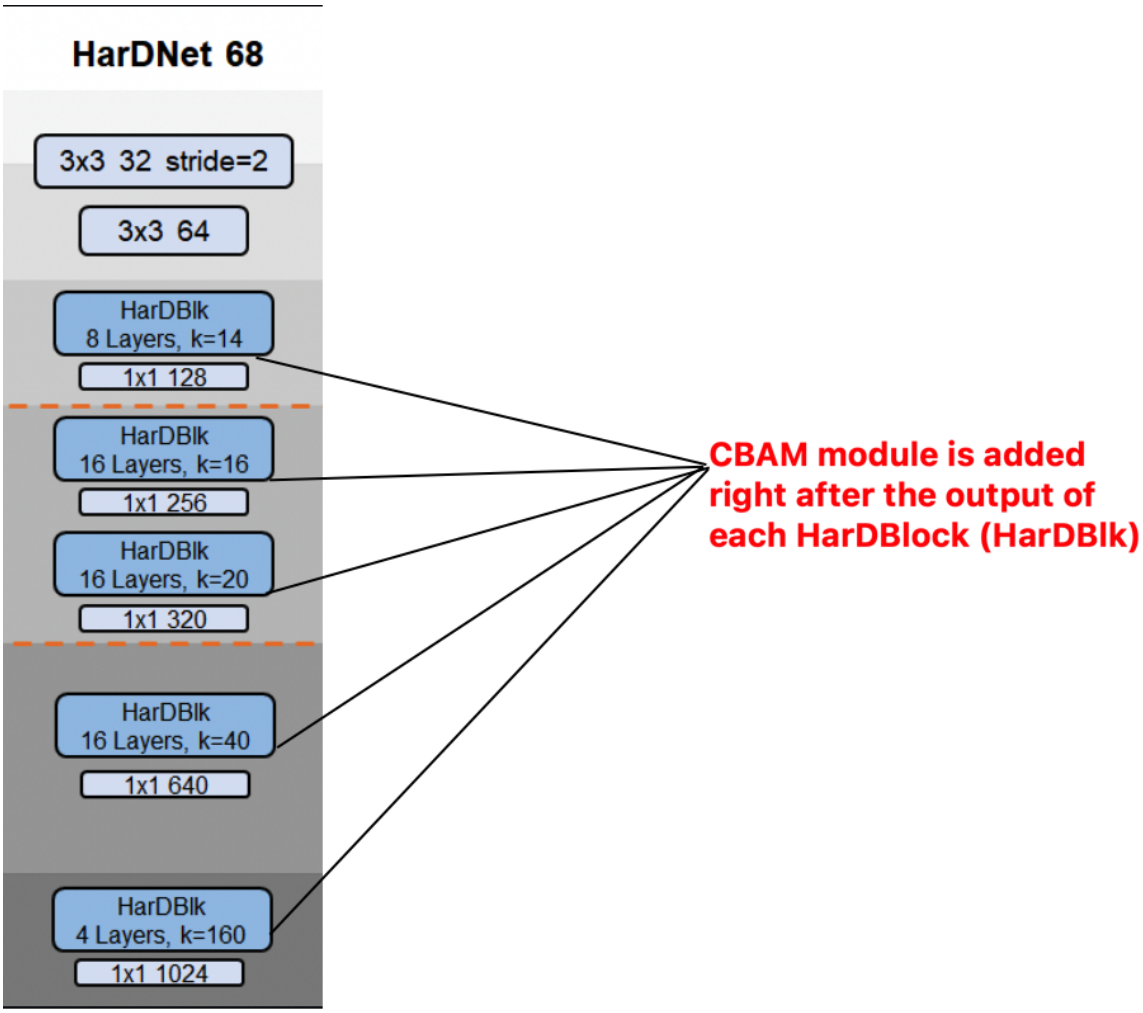

Figure 3 - Illustration of the position where the CBAM module is added with HarDNet68.

The attention-enhanced depth-wise separable versions of HarDMSEG did yields better segmentation result on polyp segmentation datasets with some trade-off in its

inference speed. The mean dice measure and the inference speed of these depth-wise separable HarDMSEG models is shown in Table 6 and Table 7 below, respectively.

Baseline of HarDMSEG	CVC-300	CVC-ClinicDB	Kvasir-SEG	CVC-ColonDB	ETIS-LaribPolypDB
HarDNet39ds	0.859	**0.896**	**0.891**	**0.723**	0.641
HarDNet39ds + CBAM	**0.873**	0.885	**0.891**	0.722	**0.673**
HarDNet68ds	0.853	0.894	0.891	**0.735**	**0.693**
HarDNet68ds + CBAM	**0.894**	**0.898**	**0.895**	0.732	0.688

Table 5 - mDice measure of attention-enhanced depth-wise separable HarDMSEG with purely depth-wise separable baselines.

Baseline of HarDMSEG	Running time in FPS
HarDNet39ds	677.7053
HarDNet39ds + CBAM	571.3972
HarDNet68ds	483.5103
HarDNet68ds + CBAM	416.4007

Table 6 - Inference speed of attention-enhanced depth-wise separable HarDMSEG with purely depth-wise separable baselines on 352x352 images.

As shown in Table 7, the inference speed of attention-enhanced depth-wise separable HarDMSEG is slower than purely depth-wise separable baseline versions of

modified HarDMSEG. However, the added attention module does help improve the effectiveness of HarDMSEG with HarDNet39ds and HarDNet68ds baselines. For the version of HarDMSEG with HarDNet39ds baseline, adding the CBAM module improves the mean dice measure on CVC-300 and ETIS-LaribPolypDB datasets. For the version with HarDNet68ds baseline, the added CBAM improves the mean dice on CVC-300 dataset by a difference of 0.041 along with marginal improvements on CVC-ClinicDB and Kvasir-SEG. Interestingly, the mean dice measure of the two attention-enhanced models in Table 6 decreases by a marginal value (*e.g.*, by a difference of 0.011 between purely depth-wise separable and attention-enhanced HarDNet39ds baselines) on some datasets. One possible explanation for this is that these modified HarDMSEG models with depth-wise separable baselines has reached a point of convergence on those datasets. Nevertheless, this experiment suggests that modifying the original HarDMSEG model architecture could be a promising direction to introduce a new method with a more balanced speed-accuracy trade-off.

References

[1] H.-Y. W. a. Y.-L. L. Chien-Hsiang Huang, "HarDNet-MSEG: A Simple Encoder-Decoder Polyp Segmentation Neural Network that Achieves over 0.9 Mean Dice and 86 FPS," *arXiv:2101.07172*, 2021.

[2] H. L. a. D. K. Taehun Kim, "UACANet: Uncertainty Augmented Context Attention for Polyp Segmentation," *arXiv:2107.02368*, 2021.

[3] D.-P. a. J. G.-P. a. Z. T. a. C. G. a. F. H. a. S. J. a. S. L. Fan, "PraNet: Parallel Reverse Attention Network for Polyp Segmentation," *MICCAI*, 2020.

[4] J. a. S. E. a. D. T. Long, "Fully convolutional networks for semantic segmentation," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[5] P. F. T. B. Olaf Ronneberger, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *Medical Image Computing and Computer-Assisted Intervention*, 2015.

[6] L.-C. a. P. G. a. K. I. a. M. K. a. Y. A. L. Chen, "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834-848, 2018.

[7] P. H. S. M. A. R. P. H. T. d. L. D. J. a. H. D. J. Debesh Jha, "Kvasir-seg: A segmented polyp dataset," in *International Conference on Multimedia Modeling*, 2020.

[8] J. S. F. J. F.-E. G. G. D. R. C. a. V. F. Bernal, "WM-DOVA maps for accurate polyp highlighting in colonoscopy: Validation vs. saliency maps from physicians," *Computerized Medical Imaging and Graphics*, vol. 43, pp. 99-111, 2015.

[9] F. J. S. a. F. V. Jorge Bernal, "Towards Automatic Polyp Detection with a Polyp Appearance Model," *Pattern Recognition*, vol. 45, no. 9, p. 3166–3182, 2012.

[10] J. T. N. S. F. M. B. C. H. Y. L. A. Q. R. O. R. B. B. I. P. K. C. S. D. Q. M.-H. L. S. S. S. D. B. P. C. H. S.-M. C. Bernal, "Comparative Validation of Polyp Detection Methods in Video Colonoscopy: Results from the MICCAI 2015 Endoscopic Vision Challenge," *IEEE Transactions on Medical Imaging*, vol. 36, no. 6, pp. 1231-1249, 2017.

[11] P. a. M. H. a. V. N. a. D. J. a. H. L. a. H. A. Chlap, "A review of medical image data augmentation techniques for deep learning applications," *Journal of Medical Imaging and Radiation Oncology*, vol. 65, no. 5, pp. 545-563, 2021.

[12] w. (https://stats.stackexchange.com/users/159052/willem), "F1/Dice-Score vs IoU," [Online]. Available: https://stats.stackexchange.com/q/276144.

[13] P. a. K. C.-Y. a. R. Y. a. H. C.-H. a. L. Y.-L. Chao, "HarDNet: A Low Memory Traffic Network," in *International Conference on Computer Vision (ICCV)*, 2019.

[14] A. a. S. N. a. P. N. a. U. J. a. J. L. a. G. A. N. a. K. Ł. a. P. I. Vaswani, "Attention is All you Need," in *Advances in Neural Information Processing Systems*, 2017.

[15] S. a. P. J. a. L. J.-Y. a. K. I. S. Woo, "CBAM: Convolutional Block Attention Module," in *European Conference on Computer Vision (ECCV)*, 2018.

[16] Y. a. B. L. a. B. Y. a. H. P. Lecun, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278-2324, 1998.

[17] I. S. G. E. H. Alex Krizhevsky, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[18] K. S. a. A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *CoRR*, 2015.

[19] W. L. Y. J. P. S. S. R. D. A. D. E. V. V. a. A. R. Christian Szegedy, "Going Deeper with Convolutions," in *Computer Vision and Pattern Recognition (CVPR)*, 2015.

[20] S. I. a. V. V. Christian Szegedy, "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning," *arXiv:1602.07261*, 2016.

[21] V. V. S. I. J. S. a. Z. W. Christian Szegedy, "Rethinking the Inception Architecture for Computer Vision," *arXiv:1512.00567*.

[22] K. a. Z. X. a. R. S. a. S. J. He, "Deep Residual Learning for Image Recognition," in *Computer Vision and Pattern Recognition (CVPR)*, 2016.

[23] X. Z. S. R. a. J. S. Kaiming He, "Identity Mappings in Deep Residual Networks," *arXiv:1603.05027*, 2016.

[24] M. a. L. Q. Tan, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," in *International Conference on Machine Learning*, 2019.

[25] A. a. B. L. a. K. A. a. W. D. a. Z. X. a. U. T. a. D. M. a. M. M. a. H. G. a. G. S. a. U. J. a. H. N. ovitskiy, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," in *International Conference on Learning Representations (ICLR)*, 2021.