MingleNet: A Novel Dual Stacking Approach for Medical Image Segmentation*

Dwayne Reinaldy[a], Hanson Gabriel Cendana[a], Bao Wei Chiam[a], Jiunn Wei Chiang[a], Vincensius Hobart Wijaya[a] and Kuan Y. Chang[a]

[a]National Taiwan Ocean University, Department of Computer Science and Engineering, No.2, Beining Rd, Keelung City, 20231, Taiwan (R.O.C.)

ARTICLE INFO

Keywords:
Medical Image Segmentation
Ensemble Learning
Dynamic Data Augmentation
Polyp Segmentation
Multiple Stacking

ABSTRACT

Medical image segmentation holds significant importance in disease diagnosis and the formulation of treatment strategies. Ensemble learning, which combines multiple models or predictions, can improve accuracy and performance in medical image segmentation. We propose MingleNet, which uses multiple layers of ensemble learning. MingleNet uses double-stacking of models, such as DoubleU-Net, DeepLabv3+, U-Net, and DeepLab, to produce masks. The first layer's masks are averaged and concatenated with the original images for the second layer. We also apply dynamic data augmentation to enhance model performances. We evaluate MingleNet on polyp segmentation benchmark datasets: Kvasir-SEG, CVC-ClinicDB, and CVC-ColonDB. On Kvasir-SEG, MingleNet achieves 93.19 Dice, 87.24% IoU, 94.15% precision, 92.25% recall, and 97.87% accuracy. On CVC-ClinicDB, MingleNet achieves 95.99% Dice, 92.29% IoU, 96.08% precision, 95.90% recall, and 99.21% accuracy. On CVC-ColonDB, MingleNet achieves 94.33% Dice, 89.28% IoU, 95.89% precision, 92.83% recall, and 99.10% accuracy. Our proposed method demonstrated competitive performance across the Kvasir-SEG, CVC-ClinicDB, and CVC-ColonDB datasets. On the CVC-ClinicDB and CVC-ColonDB benchmarks, MingleNet ranks 1st in Dice and IoU. Moreover, MingleNet ranks 8th in Dice and 17th in IoU on the Kvasir-SEG benchmark.

1. Introduction

Medical image segmentation serves the critical purpose of enhancing the visibility of anatomical structures within images, which enables more precise analysis and diagnosis [1]. By partitioning the image into different regions, it allows for extraction of relevant features and improves diagnostic accuracy. With the advancements in imaging technologies in recent years, like magnetic resonance imaging (MRI) scans and computed tomography (CT) scans, medical image segmentation has gained huge interest from researchers and practitioners.

The goal of medical image segmentation is to accurately describe the boundaries and contours of organs, tissues, or other structures of interest within medical images. The purpose of this process is to extract quantitative information, volumetric measurements, and spatial relationships, which are vital for clinical decision-making and precise intervention.

A wide range of segmentation techniques have been developed and employed in the field of medical imaging. Traditional methods include thresholding, region-based methods, clustering, and edge detection [1, 2, 3]. However, these methods frequently encounter challenges in dealing with the variability of medical images, often resulting in suboptimal outcomes.

In recent years, medical image segmentation has been revolutionized by the rise of deep learning-based techniques, with convolutional neural networks (CNNs) playing a pivotal role in this transformation. CNNs have shown exceptional performance in various segmentation tasks by automatically learning hierarchical representations and capturing contextual information from medical images. Models such as U-Net[4], SegNet[5], DeepLab[6], TransUNet [7], UNet++[8] , FCN-Transformer[9],DUCK-Net[10] and Meta-Polyp[11] have been extensively used and adapted for medical image segmentation. U-Net[4], SegNet[5], DeepLab[6], UNet++[8], DeepLabv3+[12], and DoubleU-Net[13] are all deep CNN models for semantic segmentation, featuring an encoder (compression) path to capture context and a decoder (expansion) path for precise localization. From the encoder to the decoder, U-Net[4] transfers the entire feature maps, while SegNet[5] transfers only the pooling indices. U-Net uses skip connections to fuse low-level and high-level features, and up-convolutions to upsample the feature maps[4]. DeepLab uses atrous convolutions to enlarge the receptive field of the feature maps, and atrous spatial pyramid pooling to capture multi-scale information[6]. UNet++ is an improved version of U-Net, which uses nested and dense skip connections to enhance the feature fusion between the encoder and the decoder.[8] Instead, TransUNet is a transformer-based neural network that combines the strengths of both transformers and U-Net [7]. DeepLabv3+ is an advanced iteration of DeepLab featuring spatial attention mechanisms to enhance focus on relevant regions and improve localization[12]. DoubleU-Net is an extension of the original U-Net architectures, designed to further enhance segmentation performance by incorporating dual encoding and decoding pathways to capture more intricate features[13].FCN-Transformer combines the strengths of fully convolutional networks (FCNs) and transformers to capture fine-grained details and long-range relationships within the image[9].DUCK-Net is a convolutional neural

ORCID(s):

network architecture design for accurate polyp segmentation.It can achieve excellent performance even with limited training data by using custom DUCK Block and residual downsampling[10].Meta-Polyp is also a model that focuses on poly segmentation. It combine Meta-Former architecture with UNet and utilizes a new Convformer block to capture both fine details and global context for accurate yet efficient polyp detection[11].

Ensemble learning is a machine learning technique that integrates the predictions of multiple models to improve a single model by reducing variance and exploiting the advantages of each model [14]. One of the most popular methods is Stacking. Stacking trains multiple models (usually of different types) on the same data and then uses another model, which is called a meta-learner, to learn how to best combine their predictions [15].

In this study, we explored whether integrating ensemble learning techniques could substantially improve the performance of medical image segmentation models, especially in polyp segmentation. To choose our base model, we carefully reviewed the existing literature, focusing on benchmark results such as Kvasir-SEG[16], CVC-ClinicDB [17], and CVC-ColonDB[18] in paperswithcode benchmark. We opted for DeepLabv3+[12] and DoubleU-Net[13] specifically because both have high performance in segmentation tasks and both models are the best models derived from two different models, U-Net[4] and DeepLab[6]. Therefore, we have thought about using both uniqueness to our advantage by using ensemble learning to increase performance. By using those models we leverage the diverse architectures and unique strengths of each model by capturing complimentary patterns and robustness on diverse datasets.

2. Related Works

2.1. DivergentNets

DivergentNets is a medical image segmentation technique that uses an ensemble of multiple high-performing image segmentation architectures [19]. The model combines the TriUNet segmentation model with an ensemble of well-known segmentation models, namely UNet++, FPN, DeepLabv3, and DeepLabv3+ [19]. The TriUNet model takes a single image as input and then passes it into two distinct U-Net models [19]. The outputs of these two models are combined and then processed through a third U-Net model to generate the final segmentation masks [19].

In the EndoCV2021 challenge, the TriUNet architecture used in DivergentNets was the winning model in terms of segmentation accuracy and generalization [19]. In contrast to the original TriUNet method, DivergentNets employs an ensemble of five intermediate models (TriUnet, UNet++, FPN, DeepLabv3, and DeepLabv3+), which are trained separately and then combined by averaging the pixels between each mask[19].

In comparison, DivergentNets employs five different medical image segmentation models, while MingleNet uses four models, DeepLabv3+[12], DoubleU-Net[13], DeepLab[6], and U-Net[4]. Furthermore, MingleNet employs a stacking approach with multiple layers of models, utilizing two layers of deep learning models instead of the single layer used in DivergentNets. We employ the identical method, specifically the averaging technique to combine predictions.

3. Material and Methods

3.1. Polyp Image Datasets

3.1.1. Kvasir-SEG

The Kvasir-SEG v2 dataset [16] which consists of 1000 gastrointestinal segmented polyp images and corresponding segmentation masks, was manually annotated and verified by medical experts. The image resolution ranges from 332 x 487 to 1920 x 1072 pixels.

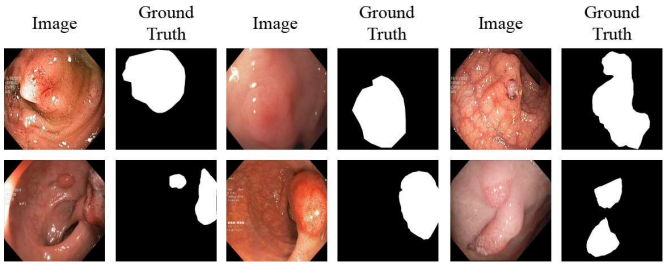

Figure 1: Example of Images and Masks from Kvasir-SEG. Each of the six images has a mask on its right side.

3.1.2. CVC-ClinicDB

CVC-ClinicDB [17] is an open-access dataset of 612 gastrointestinal segmented polyp images and their ground truth from 31 colonoscopy videos at a resolution of 384x288 pixels.

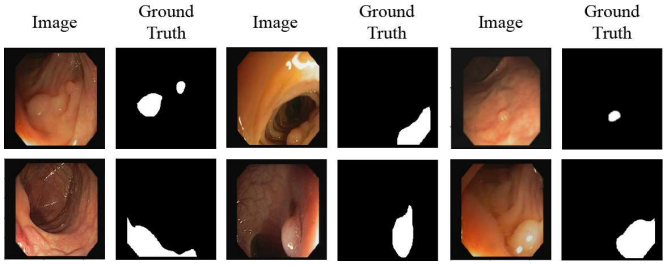

Figure 2: Example of Images and Masks from CVC-ClinicDB. Each of the six images has a mask on its right side.

3.1.3. CVC-ColonDB

CVC-ColonDB [18] is an open-access dataset of 380 colon-segmented polyp images and masks from 13 patients' colonoscopy videos at 500 x 574 pixels.

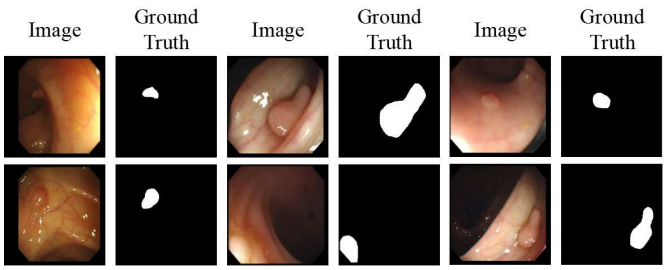

Figure 3: Example of Images and Masks from CVC-ColonDB. Each of the six images has a mask on its right side.

In this study, we rescaled Kvasir-SEG and CVC-ColonDB images to 352 x 352 pixels and CVC-ClinicDB images to 256 x 256 pixels.

3.2. Deep Learning Models for Image Segmentation

3.2.1. DeepLabv3+

DeepLab is a semantic segmentation architecture that utilizes dilated convolutions to process the input image. The output is then fine-tuned by passing it through the fully connected CRF and bilinear interpolation. DeepLabv3+ incorporates semantic information from its encoder module and recovers precise object boundaries through the decoder module. DeepLabv3+ is enriched with semantic information from the encoder module, while the decoder module effectively recovers detailed object boundaries. The encoder module allows for the extraction of features at any resolution by using atrous convolution. The output of DeepLabv3 encodes ample semantic information, while atrous convolution offers control over the density of encoder features. The decoder module facilitates the precise recovery of object boundaries. DeepLabv3+ also explores this operation and shows improvement in terms of both speed and accuracy by adapting the ResNet101 model [12].

3.2.2. DoubleU-Net

DoubleU-Net is a semantic segmentation model consisting of two U-Net architectures. It contains five components: two U-Net networks, VGG19, a squeeze-and-excite block, and atrous spatial pyramid pooling (ASPP). The first U-Net utilizes VGG19 as an encoder and ASPP as the input of the decoder. The squeeze-and-excite block improves the feature maps in the encoder of the first U-Net and those of both decoders. The output of the first U-Net and the corresponding image's mask are multiplied by the input pictures. The second U-Net also utilizes ASPP and the squeeze-and-excite block. Finally, the result is obtained through concatenation of the outputs from both U-Nets. DoubleU-Net strengthens the segmentation process by employing two U-Nets [13].

3.3. Comparison between DeepLabv3+ and DoubleU-Net

DeepLabv3+[12] emphasizes the use of atrous convolutions for capturing multiscale contextual information. It is renowned for its exceptional performance in segmentation tasks, particularly in situations where contextual information is essential. DoubleU-Net[13] expands upon the U-Net framework by incorporating a dual pathway for information exchange. Its primary objective is to fuse features across multiple scales, enhancing its ability to handle objects of various sizes. DoubleU-Net[13] and DeepLabv3+[12] both utilize ASPP (Atrous Spatial Pyramid Pooling) as part of their architecture. Moreover, these models are also utilizing a backbone in their architecture (DoubleU-Net[13] uses VGG19 as a backbone, while DeepLabv3+[12] uses ResNet101 as a backbone).

3.4. Concept of Ensemble Learning

Ensemble learning is a powerful machine learning approach that involves combining multiple individual models to achieve more accurate predictions or classifications than what a single model can achieve on its own[15]. The fundamental idea behind ensemble learning is to leverage the diversity of these models to overcome the limitations and biases inherent in individual models, thereby enhancing overall performance and generalization capabilities[15].

3.4.1. Stacking

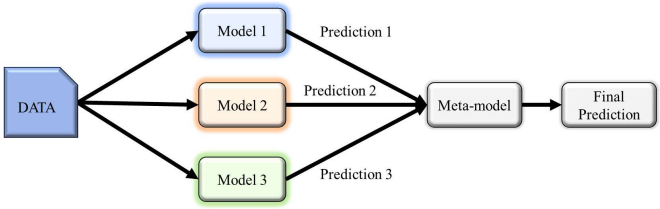

Figure 4: Concept of Stacking

Stacking is an ensemble learning technique that improves predictive performance by combining multiple models. It uses their predictions as inputs to a meta-model, which makes the final prediction. The meta-model compensates for the weaknesses of individual base models, leading to better performance than using a single base model [15].

3.5. MingleNet Architecture

MingleNet is a convolutional neural network (CNN) architecture that fuses the incorporation of multiple layers of models, this architecture is illustrated in the accompanying Fig. 5. In the architecture of MingleNet, the first layer consists of two different deep learning models, namely DeepLabv3+[12] and DoubleU-Net[11]. The first layer model takes the original image with three channels, Red, Green, and Blue(RGB) as input. Subsequently, we average the output of the first layer which consists of three different output masks. After averaging, the average mask output of the first layer is concatenated with the channels of the original input image, this resulted in the image consisting of four different channels (Red, Green, Blue, and average output masks of the first layer).

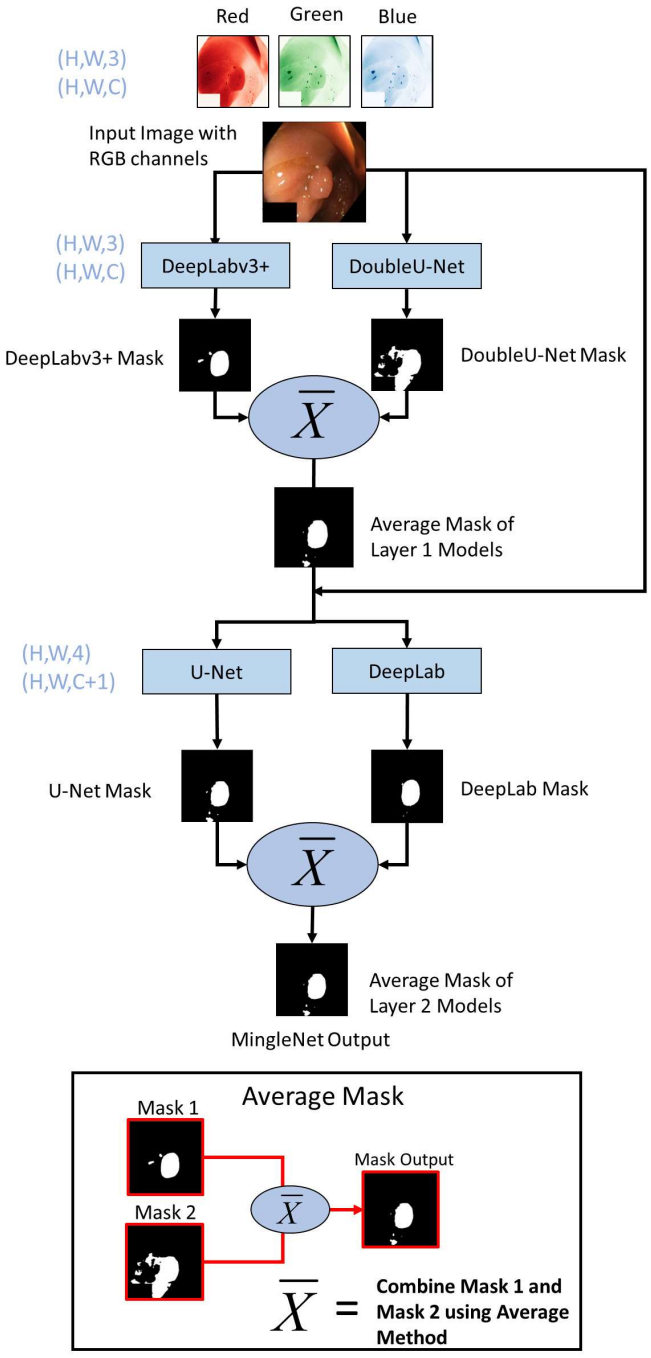

Figure 5: MingleNet architecture. Diagram of the proposed ensemble learning technique

The second layer consists of two distinct deep learning models namely U-Net[4] and DeepLab[6]. The second layer model takes the image of four different channels (Red, Green, Blue, and average output masks of the first layer) as an input. Afterward, we average the output masks of the second layer as the final output. In addition, we selected U-Net[4] and DeepLab[6] for the second layer instead of DeepLabv3+[12] and DoubleU-Net[13] because DeepLabv3+[12] employs a ResNet101 backbone, and DoubleU-Net[13] utilizes a VGG19 backbone, both of which lack support for a 4-channel input.

We conducted training for 600 epochs on each model in the first layer, employing a batch size of 4, and saved the best-performing model. Furthermore, we opted for AdamW Optimizer with a learning rate of 0.0001, weight decay of 0.004, and binary crossentropy loss as the objective function for the first layer. Additionally, for the first layer, we conducted dynamic data augmentation for every epoch.

For the second layer, we conducted training for 200 epochs on each model, employing a batch size of 4. In addition, we chose the AdamW Optimizer and binary crossentropy loss as the objective function, concurrently lowering the learning rate to 0.00001 with a weight decay of 0.004. This adjustment aims to minimize the likelihood of the models bypassing optimal parameters, facilitating a more effective settling into the minimum of the loss function. Furthermore, in the second layer, we opted not to use dynamic data augmentation due to the time complexity associated with its implementation. The dynamic data augmentation process requires loading, predicting, and averaging first-layer models for each epoch before concatenating them with the RGB input for the second layer which requires a significant amount of time.

3.6. Implementation Details

We experimented on an Intel Core i7-12700 processor with an RTX 3080 TI GPU and 32GB of RAM at the Biomedical AI Lab at National Taiwan Ocean University to run Tensorflow 2.14.0 and Python 3.10.13.

3.7. Dynamic Image Augmentation

We adapted the image augmentation technique proposed by DUCK-Net [10], which builds upon the method introduced by Sanderson and Matuszewski [20]. Our novel approach, called Dynamic Image Augmentation, generates fresh augmented images before each training epoch. We implemented this method using the Albumentations library [21]. Specifically, we enhanced the DUCK-Net method by introducing randomized horizontal and vertical flips and Gaussian blur, all with a 0.5 probability.

1. Horizontal flips and vertical flips (with a probability of 0.5),
2. Color jitter is applied with a brightness range of [0.6, 1.6], a fixed contrast of 0.2, a saturation factor of 0.1, and a hue factor of 0.01,
3. Affine transformation with rotations within the range of $[-180°, 180°]$, horizontal and vertical translations within $[-0.125, 0.125]$, scaling with a magnitude within $[0.5, 1.5]$, and shearing with an angle within $[-22.5°, 22.5°]$.
4. Gaussian blur with blur limit of $[25, 25]$, sigma limit of $[0.001, 2.0]$, and probability of 0.5.

We use dynamic image augmentation to introduce diversity into our datasets. Techniques like flipping, adjusting brightness, rotation, and blurring enhance dataset variability.

By incorporating these transformations, our models become more adaptable and better equipped to handle various scenarios.

In addition, we introduce probability-based horizontal and vertical flips to enhance data variation in each epoch, improving model generalization. This means that images within an epoch can exhibit only horizontal flips, only vertical flips, both horizontal and vertical flips, or no flips at all. Regarding color jitter and affine transformations, we define specific value ranges for this data augmentation. Consequently, each epoch experiences different color jitter and affine transformations, as these parameters are randomly selected from their respective ranges. Lastly, Gaussian blur is applied probabilistically, allowing images in each epoch to exhibit either blur or no blur.

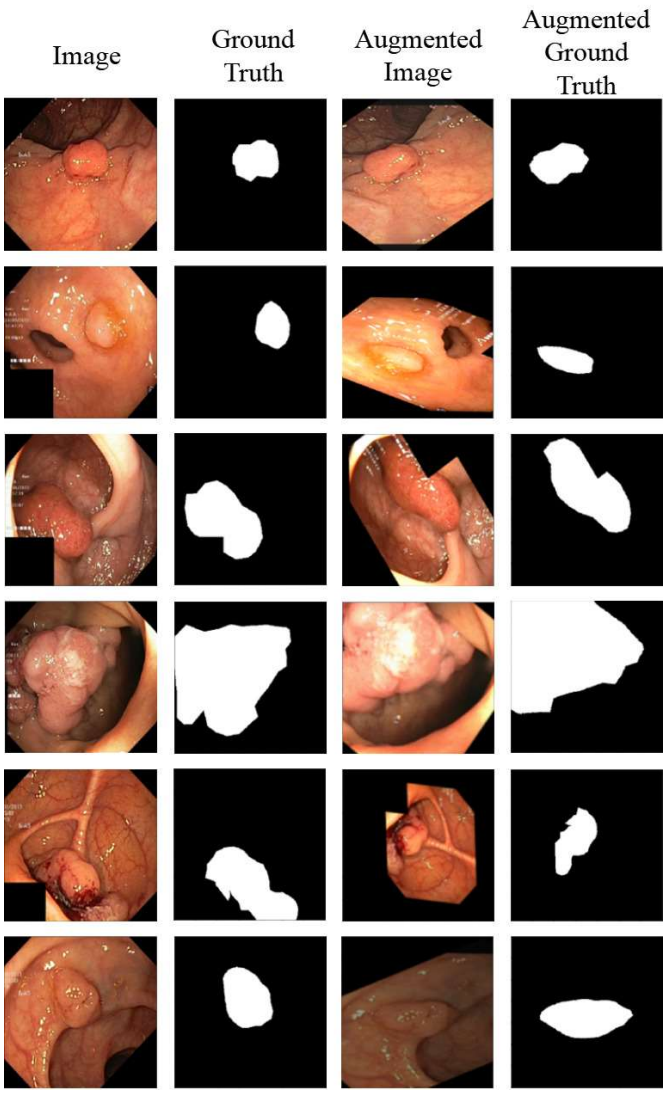

Figure 6: Example of images, masks, augmented images, and augmented masks. Each of the six images is followed by a mask, an augmented image, and its corresponding augmented mask.

3.8. Evaluation

For evaluation, we use the following performance metrics: Dice-Coefficient, IoU (Intersection over Union), Precision, Recall, and Accuracy. All these metrics are calculated based on the actual mask we have. The formula is presented below:

$$\text{DiceCoefficient} = \frac{2TP}{2TP + FP + FN} \tag{1}$$

$$\text{IoU} = \frac{TP}{TP + FP + FN} \tag{2}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{4}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \tag{5}$$

All the metrics used in this study are based on the binary classification's confusion matrix, where TP, FP, TN, and FN correspond to the true positive, false positive, true negative, and false negative, respectively [22]. These metrics ensure that our evaluation directly compares predicted results to the ground truth represented by the true mask.

4. Results

Table 1 shows that data augmentation improves all semantic segmentation models significantly. We use Dice-Coefficient and IoU as evaluation metrics because they are more important for image segmentation. Notably, the most remarkable improvement is observed in the DoubleU-Net[23] model, which experiences a notable increase in Dice by 7.06% and IoU by 11.09% after the application of DUCK-Net Data Augmentation. Furthermore, after incorporating our Data Augmentation, the DoubleU-Net model showcases a further enhancement, with Dice improving by 1.9% and IoU by 3.23% compared to DoubleU-Net model with DUCK-Net Augmentation. DeepLabv3+[12] is already a good model for the Kvasir-SEG dataset, but it still improves by 4.42% in Dice and 7.24% in IoU after applying DUCK-Net Augmentation. Moreover, after applying our Data Augmentation, DeepLabv3+ improves by 0.24% in Dice and 0.4% in IoU when compared to DeepLabv3+ with DUCK-Net Augmentation. This experiment demonstrates that dynamic data augmentation enhances the performance of each deep learning model. These improvements are particularly significant given the characteristics of the datasets involved. For example, the Kvasir-SEG dataset, with its larger size of 1000 samples, facilitates better generalization, even when some images are blurred due to augmentation. The diversity

Table 1
Performance comparison of Models with Various Data Augmentation Techniques with different data sets. The best results are in bold.

Dataset	Model	Without Data Augmentation		DUCK-Net Augmentation		MingleNet Augmentation	
		Dice	IoU	Dice	IoU	Dice	IoU
Kvasir-SEG	DeepLab[6]	75.41%	66.21%	70.91%	54.93%	**87.61%**	**77.95%**
	U-Net[4]	83.59%	74.28%	80.93%	67.98%	**86.73%**	**76.57%**
	DoubleU-Net[13]	83.62%	71.86%	90.68%	82.95%	**92.58%**	**86.18%**
	DeepLabv3+[12]	87.20%	77.30%	91.62%	84.54%	**91.86%**	**84.94%**
CVC-ClinicDB	DeepLab[6]	86.72%	74.09%	91.24%	83.89%	**93.35%**	**87.53%**
	U-Net[4]	91.06%	81.51%	92.17%	85.48%	**94.25%**	**89.13%**
	DoubleU-Net[13]	93.66%	88.08%	**97.10%**	**94.36%**	95.89%	92.11%
	DeepLabv3+[12]	93.76%	88.25%	**97.14%**	**94.44%**	95.19%	90.83%
CVC-ColonDB	DeepLab[6]	75.41%	66.21%	**91.04%**	**83.56%**	90.87%	83.28%
	U-Net[4]	86.50%	76.20%	**90.62%**	**82.86%**	90.46%	82.15%
	DoubleU-Net[13]	90.42%	82.51%	92.90%	86.75%	**93.74%**	**88.21%**
	DeepLabv3+[12]	89.60%	81.16%	**90.81%**	**83.16%**	90.37%	82.44%

Table 2
Performance of MingleNet on Kvasir-SEG. The best results are in bold.

MingleNet	Dice	IoU	Precision	Recall	Accuracy
Layer 1:DoubleU-Net	92.58%	86.18%	92.28%	**92.87%**	97.64%
Layer 1:DeepLabv3+	91.86%	84.94%	93.76%	90.03%	97.48%
Layer 2:U-Net	93.13%	87.14%	93.92%	92.35%	97.84%
Layer 2:Deeplab	92.69%	86.37%	**95.12%**	90.37%	97.74%
Final Output	**93.19%**	**87.24%**	94.15%	92.25%	**97.87%**

within this dataset ensures that essential features are still captured, despite the Gaussian blur, thereby augmenting variability in lighting and texture. Conversely, datasets like CVC-ClinicDB and CVC-ColonDB, with fewer samples (612 and 480 respectively), suffer more from the detrimental effects of Gaussian blur. In smaller datasets, preserving details is crucial for maintaining model performance, making augmentation strategies that avoid Gaussian blur more suitable. Our augmentation, which includes Gaussian blur, proves beneficial for Kvasir-SEG due to its larger sample size, where the model can compensate for information loss and benefit from increased variability. On the other hand, DuckNet Augmentation, which excludes Gaussian blur, is more effective for datasets like CVC-ClinicDB and CVC-ColonDB, where preserving fine details is paramount for effective learning from smaller datasets. This strategy provides sufficient variability without compromising the integrity of important features in the images, ensuring optimal performance in such contexts.

Table 2 presents the performance of various MingleNet components on the Kvasir-SEG[16] benchmark. DoubleU-Net[13] outperforms DeepLabv3+[12] in Dice, IoU, and Recall at the first layer. At the second layer, U-Net[4] performs better than DeepLab[6] in Dice, IoU, and Recall after averaging the predicted masks from DoubleU-Net[13] and DeepLabv3+[12]. Comparing DoubleU-Net[13] at the first layer, MingleNet's final output improves Dice by 0.61% and IoU by 1.13%.

Table 3 compares the performance of various semantic segmentation models on the Kvasir-SEG benchmark. Meta-Polyp [11] achieves the highest scores in Dice (95.90%) and IoU (87.24%). DUCK-Net [10] achieves the highest scores in Precision (96.28%), Recall (93.79%), and Accuracy (98.42%). These results show the effectiveness of the proposed MingleNet architecture in semantic segmentation tasks. Although MingleNet does not outperform models like DUCK-Net [10] and Meta-Polyp [11], it still shows robust performance for this dataset.

Fig. 7 shows the progressive result in MingleNet on CVC-ClinicDB test set. Starting from base models consisting of DoubleU-Net [13] and DeepLabv3+[12], whose predictions are averaged and fed pass into the second layer comprising [4] and DeepLab[6]. Then MingleNet Output is the result of the averaged prediction masks of U-Net[4] and DeepLab[6].

Fig. 7 reveals that the resulting output shows that MingleNet were to identify the shape of the polyp images. It also shows fewer false positives compared to the base models predictions and were able to identify the shape of the polyp objects, however results in the first row shows that

Table 3
Result comparison on Kvasir-SEG. The best results are in bold.

Model	Dice	IoU	Precision	Recall	Accuracy
U-Net[4]	83.59%	74.28%	86.45%	76.14%	93.94%
TransResU-Net(pre-trained)[24]	88.84%	82.14%	91.06%	90.22%	96.51%
FCN-Transformer(pre-trained)[9]	92.20%	85.54%	92.38%	92.03%	97.49%
DUCK-Net(pre-trained)[10]	95.02%	90.51%	**96.28%**	**93.79%**	**98.42%**
Meta-Polyp(pre-trained)[11]	**95.90%**	**92.10%**	-	-	-
MingleNet	93.19%	87.24%	94.15%	92.25%	97.87%

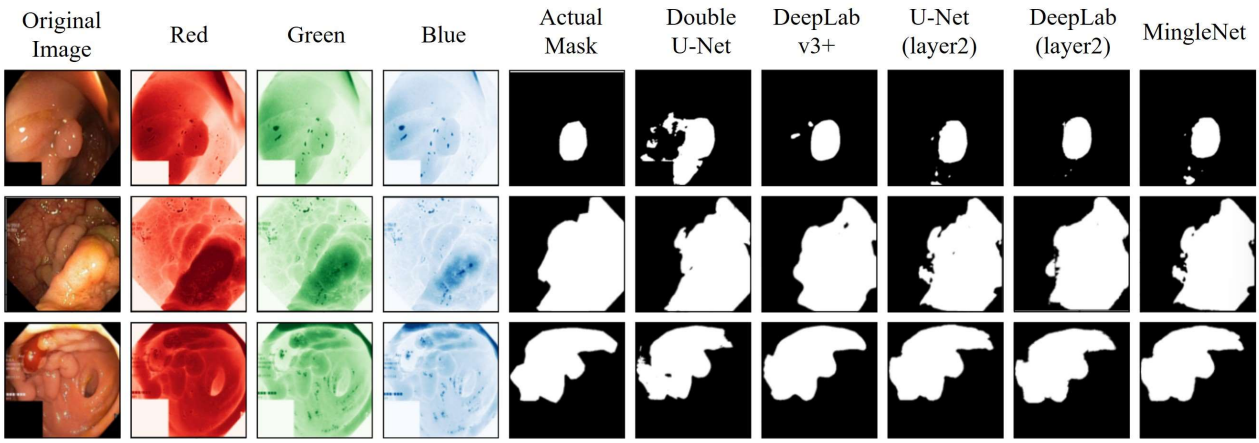

Figure 7: Segmented polyp masks of MingleNet components on Kvasir-SEG benchmark.

The figure displays the actual mask, RGB channel, and the result of each segmentation from the first layer (DoubleU-Net and DeepLabv3+), the second layer (U-Net, DeepLab), and the final output.

DeepLab[6] prediction results have shown a better visualization showing less false positives compared to MingleNet Output.

Table 4 presents the performance of various MingleNet components on the CVC-ClinicDB[17] benchmark. DoubleU-Net[13] outperforms DeepLabv3+[12] in Dice and IoU at the first layer. At the second layer, U-Net[4] performs better than DeepLab[6] in Dice, IoU, and Precision after averaging the predicted masks from DoubleU-Net[13] and DeepLabv3+[12]. Comparing DoubleU-Net[13] at the first layer, MingleNet's final output improves Dice by 0.10% and IoU by 0.19%.

Fig. 8 shows the progressive result in MingleNet on CVC-ClinicDB test set. Starting from base models consisting of DoubleU-Net[13] and DeepLabv3+[12], whose predictions are averaged and fed pass into the second layer comprising U-Net[4] and DeepLab[6]. Then MingleNet Output is the result of the averaged prediction masks of U-Net[4] and DeepLab[6]. Fig. 8 reveals that the resulting output shows that MingleNet were to identify the shape of the polyp objects. All rows show that it successfully reduced false positives that the base models predicted, improving segmentation.

Table 5 presents a comparative evaluation of various semantic segmentation models. Standing out among the evaluated models is our model MingleNet, which attains the highest scores in Dice (95.99%), IoU (92.29%), Recall (95.95%), and Accuracy (99.21%), except for Precision in which FCN-Transformer achieves the highest with a score of 96.59%. These results underscore the efficacy of the proposed MingleNet architecture in semantic segmentation tasks, outperforming other segmentation models in Dice and IoU.

Table 6 presents the performance of various MingleNet components on the CVC-ColonDB [18] benchmark. DoubleU-Net[13] outperforms DeepLabv3+[12] in Dice, IoU, and Recall at the first layer. At the second layer, DeepLab[6] performs better than U-Net[4] in Dice, IoU, and Recall after averaging the predicted masks from DoubleU-Net[13] and DeepLabv3+[12]. Comparing DoubleU-Net [13] at the first layer, MingleNet's final output improves Dice by 0.63% and IoU by 1.21%.

Fig. 9 shows the progressive result in MingleNet on CVC-ColonDB test set. Starting from base models consisting of DoubleU-Net[13] and DeepLabv3+[12], whose predictions are averaged and fed pass into the second layer comprising U-Net[4] and DeepLab[6]. Then MingleNet Output is the result of the averaged prediction masks of U-Net[4] and DeepLab[6]. Fig. 9 reveals that the resulting output shows that MingleNet was able to identify the shape of the polyp objects, although there are some issues in Row 3 where it successfully decreased its false negatives in the

Table 4
Performance of MingleNet on CVC-ClinicDB. The best results are in bold.

MingleNet	Dice	IoU	Precision	Recall	Accuracy
Layer 1:DoubleU-Net	95.89%	92.11%	95.61%	**96.17%**	99.18%
Layer 1:DeepLabv3+	95.19%	90.83%	95.72%	94.67%	99.05%
Layer 2:U-Net	95.94%	92.20%	**96.17%**	95.71%	99.20%
Layer 2:Deeplab	95.56%	91.51%	94.69%	96.46%	99.11%
Final Output	**95.99%**	**92.29%**	96.08%	95.90%	**99.21%**

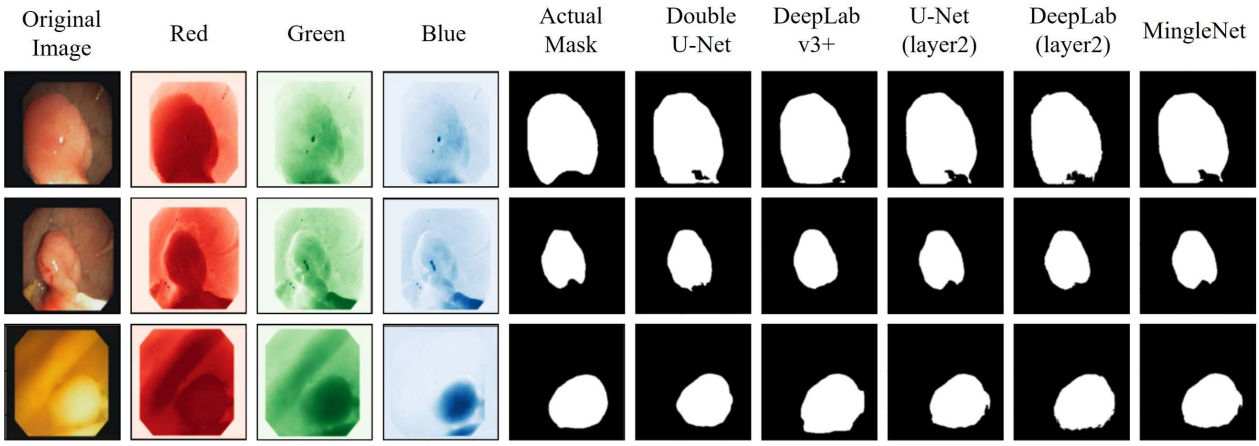

Figure 8: Segmented polyp masks of MingleNet components on the CVC-ClinicDB benchmark.

The figure displays the actual mask, RGB channel, and the result of each segmentation from the first layer (DoubleU-Net and DeepLabv3+), the second layer (U-Net, DeepLab), and the final output.

polyp object area but still showed false positives in the outer parts.

Table 7 presents a comparative evaluation of various semantic segmentation models. Standing out among the evaluated models is our model MingleNet, which attains the highest scores in Dice (94.34%), IoU (89.28%) and Precision(95.89%).While DUCK-Net[10] scores highest in Recall (93.92%) and Accuracy (99.29%). These results underscore the efficacy of the proposed MingleNet architecture in semantic segmentation tasks, outperforming other segmentation models in Dice IoU and Precision.

Table 5
Result comparison on CVC-ClinicDB. The best results are in bold.

Model	Dice	IoU	Precision	Recall	Accuracy
U-Net[4]	91.06%	81.51%	95.26%	87.22%	97.77%
FCN-Transformer(pre-trained)[9]	88.00%	78.58%	**96.59%**	80.82%	96.45%
DUCK-Net(pre-trained)[10]	94.78%	90.09%	94.68%	94.89%	99.07%
MingleNet	**95.99%**	**92.29%**	96.08%	**95.90%**	**99.21%**

Table 6
Performance of MingleNet on CVC-ColonDB. The best results are in bold.

MingleNet	Dice	IoU	Precision	Recall	Accuracy
Layer 1:DoubleU-Net	93.74%	88.21%	94.94%	92.56%	99.00%
Layer 1:DeepLabv3+	90.37%	82.44%	**96.29%**	85.14%	98.53%
Layer 2:U-Net	93.60%	87.97%	95.88%	91.42%	98.99%
Layer 2:Deeplab	93.91%	88.52%	94.74%	**93.08%**	99.02%
Final Output	**94.33%**	**89.28%**	95.89%	92.83%	**99.10%**

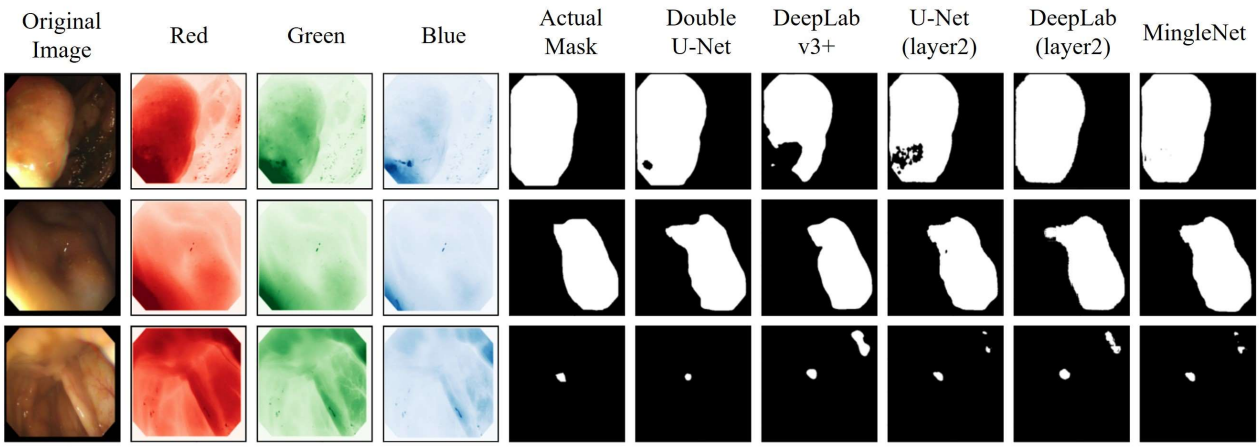

Figure 9: Segmented polyp masks of MingleNet components on CVC-ColonDB benchmark. The figure displays the actual mask, RGB channel, and the result of each segmentation from the first layer (DoubleU-Net and DeepLabv3+), the second layer (U-Net, DeepLab), and the final output.

Table 7
Result comparison on CVC-ColonDB. The best results are in bold.

Model	Dice	IoU	Precision	Recall	Accuracy
U-Net[4]	86.50%	76.2%	94.34%	79.94%	97.99%
Meta-Polyp(pre-trained)[11]	86.70%	79.00%	-	-	-
FCN-Transformer(pre-trained) [9]	90.73%	83.04%	91.07%	90.40%	98.99%
DUCK-Net (pre-trained) [10]	93.53%	87.85%	93.14%	**93.92%**	**99.29%**
MingleNet	**94.34%**	**89.28%**	**95.89%**	92.84%	99.10%

5. Discussion

This study shows the performance of ensemble learning over individual base models. Ensemble learning uses the collaborative strength of these base models, where each model adds its unique insights. This approach improves accuracy and robustness significantly. Our ensemble learning model MingleNet outperformed each base model consistently. The individual base models did well in some aspects of the task, but they had limitations in handling the full complexity of the benchmarks. Ensemble learning fills these gaps by using the diverse patterns captured by each model, leading to a better understanding of the data.

Additionally, we show that dynamic data augmentation improves the base models' performance. Dynamic data augmentation adds augmentations to the training input before each training epoch. These data variations create diverse datasets for training models. The diversity in training data enhances the model performance.

MingleNet performs well on medical image segmentation. According to paperswithcode, MingleNet ranks 8th in Dice (93.19%) and 17th in IoU (87.24%) on the Kvasir-SEG benchmark. On the CVC-ClinicDB[17] and CVC-ColonDB[18] benchmarks, MingleNet ranks 1st in Dice (95.99% and 94.34%) and IoU (92.29% and 89.28%) respectively.

This study shows that using weaker models at the second layer (U-Net[4] and DeepLab[6], weaker than DoubleU-Net[13] and DeepLabv3+[12]) improves the first layer's result slightly. This is because the second layer models use the average of the first layer as the fourth channel. The fourth channel acts as a hint for the second layer models. Moreover, averaging deep learning models improves their accuracy. This is because each model has its own strengths and weaknesses, and averaging them can overcome their weaknesses.

References

[1] R. Wang, T. Lei, R. Cui, B. Zhang, H. Meng, A. K. Nandi, Medical image segmentation using deep learning: A survey, IET Image Processing 16 (5) (2022) 1243–1267. doi:https://doi.org/10.1049/ipr2.12419.

[2] R. M. Haralick, L. G. Shapiro, Image segmentation techniques, Computer Vision, Graphics, and Image Processing 29 (1) (1985) 100–132. doi:https://doi.org/10.1016/S0734-189X(85)90153-7.

[3] D. L. Pham, C. Xu, J. L. Prince, Current methods in medical image segmentation, Annual Review of Biomedical Engineering 2 (1) (2000) 315–337, pMID: 11701515. doi:10.1146/annurev.bioeng.2.1.315.

[4] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: N. Navab, J. Hornegger, W. M. Wells, A. F. Frangi (Eds.), Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, Springer International Publishing, Cham, 2015, pp. 234–241.

[5] V. Badrinarayanan, A. Kendall, R. Cipolla, Segnet: A deep convolutional encoder-decoder architecture for image segmentation, IEEE

Transactions on Pattern Analysis and Machine Intelligence 39 (12) (2017) 2481–2495. doi:10.1109/TPAMI.2016.2644615.

[6] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A. L. Yuille, Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs, IEEE Transactions on Pattern Analysis and Machine Intelligence 40 (4) (2018) 834–848. doi:10.1109/TPAMI.2017.2699184.

[7] J. Chen, Y. Lu, Q. Yu, X. Luo, E. Adeli, Y. Wang, L. Lu, A. L. Yuille, Y. Zhou, Transunet: Transformers make strong encoders for medical image segmentation, arXiv preprint arXiv:2102.04306 (2021).

[8] Z. Zhou, M. M. Rahman Siddiquee, N. Tajbakhsh, J. Liang, Unet++: A nested u-net architecture for medical image segmentation, in: D. Stoyanov, Z. Taylor, G. Carneiro, T. Syeda-Mahmood, A. Martel, L. Maier-Hein, J. M. R. Tavares, A. Bradley, J. P. Papa, V. Belagiannis, J. C. Nascimento, Z. Lu, S. Conjeti, M. Moradi, H. Greenspan, A. Madabhushi (Eds.), Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support, Springer International Publishing, Cham, 2018, pp. 3–11.

[9] E. Sanderson, B. J. Matuszewski, Fcn-transformer feature fusion for polyp segmentation, in: G. Yang, A. Aviles-Rivero, M. Roberts, C.-B. Schönlieb (Eds.), Medical Image Understanding and Analysis, Springer International Publishing, Cham, 2022, pp. 892–907.

[10] R.-G. Dumitru, D. Peteleaza, C. Craciun, Using duck-net for polyp image segmentation, Scientific Reports 13 (1) (2023) 9803. doi:10.1038/s41598-023-36940-5.

[11] Q. Trinh, Meta-polyp: A baseline for efficient polyp segmentation, in: 2023 IEEE 36th International Symposium on Computer-Based Medical Systems (CBMS), IEEE Computer Society, Los Alamitos, CA, USA, 2023, pp. 742–747. doi:10.1109/CBMS58004.2023.00312. URL https://doi.ieeecomputersociety.org/10.1109/CBMS58004.2023.00312

[12] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, H. Adam, Encoder-decoder with atrous separable convolution for semantic image segmentation, in: V. Ferrari, M. Hebert, C. Sminchisescu, Y. Weiss (Eds.), Computer Vision – ECCV 2018, Springer International Publishing, Cham, 2018, pp. 833–851.

[13] D. Jha, M. A. Riegler, D. Johansen, P. Halvorsen, H. D. Johansen, Doubleu-net: A deep convolutional neural network for medical image segmentation, in: 2020 IEEE 33rd International symposium on computer-based medical systems (CBMS), IEEE, 2020, pp. 558–564.

[14] D. Opitz, R. Maclin, Popular ensemble methods: An empirical study, Journal of artificial intelligence research 11 (1999) 169–198.

[15] P. Proskura, A. Zaytsev, Effective training-time stacking for ensembling of deep neural networks, in: Proceedings of the 2022 5th International Conference on Artificial Intelligence and Pattern Recognition, AIPR '22, Association for Computing Machinery, New York, NY, USA, 2023, p. 78–82. doi:10.1145/3573942.3573954.

[16] D. Jha, P. H. Smedsrud, M. A. Riegler, P. Halvorsen, T. de Lange, D. Johansen, H. D. Johansen, Kvasir-seg: A segmented polyp dataset, in: Y. M. Ro, W.-H. Cheng, J. Kim, W.-T. Chu, P. Cui, J.-W. Choi, M.-C. Hu, W. De Neve (Eds.), MultiMedia Modeling, Springer International Publishing, Cham, 2020, pp. 451–462.

[17] J. Bernal, F. J. Sánchez, G. Fernández-Esparrach, D. Gil, C. Rodríguez, F. Vilariño, Wm-dova maps for accurate polyp highlighting in colonoscopy: Validation vs. saliency maps from physicians, Computerized Medical Imaging and Graphics 43 (2015) 99–111. doi:https://doi.org/10.1016/j.compmedimag.2015.02.007.

[18] D. Vázquez, J. Bernal, F. J. Sánchez, G. Fernández-Esparrach, A. M. López, A. Romero, M. Drozdzal, A. Courville, A benchmark for endoluminal scene segmentation of colonoscopy images, Journal of Healthcare Engineering 2017 (2017) 4037190. doi:10.1155/2017/4037190.

[19] V. L. Thambawita, S. Hicks, P. Halvorsen, M. Riegler, Divergentnets:medical image segmentation by network ensemble, ArXiv abs/2107.00283 (2021).

[20] E. Sanderson, B. J. Matuszewski, Fcn-transformer feature fusion for polyp segmentation, in: G. Yang, A. Aviles-Rivero, M. Roberts, C.-B. Schönlieb (Eds.), Medical Image Understanding and Analysis, Springer International Publishing, Cham, 2022, pp. 892–907.

[21] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, A. A. Kalinin, Albumentations: Fast and flexible image augmentations, Information 11 (2) (2020). doi:10.3390/info11020125.

[22] J. Lever, M. Krzywinski, N. Altman, Classification evaluation, Nature Methods 13 (8) (2016) 603–604. doi:10.1038/nmeth.3945.

[23] N. K. Tomar, D. Jha, S. Ali, H. D. Johansen, D. Johansen, M. A. Riegler, P. Halvorsen, Ddanet: Dual decoder attention network for automatic polyp segmentation, in: Pattern Recognition. ICPR International Workshops and Challenges: Virtual Event, January 10-15, 2021, Proceedings, Part VIII, Springer, 2021, pp. 307–314.

[24] N. K. Tomar, A. Shergill, B. Rieders, D. Jha, Transresu-net: A transformer based resu-net for real-time colon polyp segmentation, Vol. 2023, 2023, pp. 1–4. doi:10.1109/EMBC40787.2023.10340572.