Human segmentation using U-Net architecture

Amirmohammad Shahbandegan, Computer Science Department, Lakehead University

I. Introduction

The aim of this assignment is to implement a U-Net neural network to detect and segment humans. The desired output is a mask with the same size as the original input where the pixels corresponding to a human in the original image are activated.

A. Dataset

The data is composed of 300 images containing human subjects and a mask image for each input where the desired area is specified with white color. A few sample images from the dataset and their corresponding masks are shown in Fig. 1

Fig. 1. Sample images drawn from the dataset and their corresponding masks.

Fig. 2. Sample images drawn from the dataset with data augmentation and their corresponding masks.

B. Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are a class of deep learning models most commonly used in image processing tasks. These networks gained a high popularity in the past decade for their ability in reducing the number of network parameters [1]. The most important element of a convolutional neural network is the convolution layer. The convolution for one pixel in the next layer is calculated according to Formula 1

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(m,n)K(i-m, j-n) \tag{1}$$

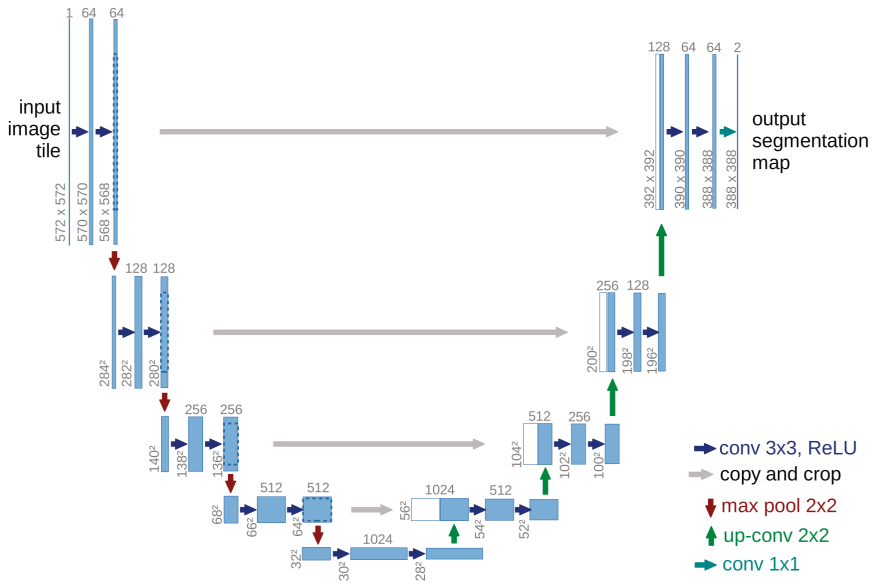

Fig. 3. U-Net network architecture with convolution, pooling, and up-sampling layers [2]

TABLE I
PERFORMANCE OF TRAINED MODELS

	Optimizer	Initial Learning Rate	Epochs	Data Augmentation	Early Stopping	Training Accuracy	Testing Accuracy
Model A	adam	10^{-4}	100	No	No	94.65%	86.42%
Model B	adam	10^{-4}	100	Yes	No	88.41%	87.77%
Model C	adam	10^{-4}	100	Yes	Yes	86.37%	**88.05%**

C. U-Net Architecture

U-Net [2] is a deep CNN architecture proposed by Ronneberger et. al. in 2015 for biomedical image segmentation and found great popularity for other segmentation tasks later on. The architecture of this network is depicted in Fig. 3. It consists of convolution, pooling and up-conv layers. The input is first encoded to a latent space using convolutional and pooling operations and then this encoding is used to recunstruct the mask image using convolution and up-conv operations. The model also benefits from skip connections to further improve the decoding procedure.

II. METHODOLOGY

To asses the quality of the designed network based on the U-Net architecture, the dataset is randomly split into two groups, 70% for training and 30% for testing. Among the training data, 20% has been chosen for model validation. Three different models were experimented to analyze the effect of regularization methods such as data augmentation and early stopping.

The first model (Model A) follows the basic U-Net model as explained in Section I. Second model (Model B), adds five data augmentation layers before the first layer of model A. These augmentation layers are used in this model: random flip, random rotation, random zoom, and random contrast. Since we are dealing with an image segmentation task, we need to apply the same augmentations to the labels(masks) as well. Fig. 2 depicts a sample output of the augmentation method and how the same pipline is applied to the masks. The third model (Model C) is based on Model B. Model C utilizes early stopping with a patience of 10 epochs to avoid overfitting whereas model B does not. The total number of trainable parameters in all three models is $31,032,837$. The optimizer used in this work is the Adam optimizer [3] with an initial learning rate of 10^{-4}.

The models are implemented using Python programming language and Tensorflow library on Google Colaboratory environment. The code used to run the experiments are submitted along with this report.

III. EXPERIMENTAL RESULTS

All three models are trained with the same training data for 100 epochs and the accuracy of the models are then calculated using a common testing set so that the results are comparable. A summary of the performance of the models and their respective hyperparameters are shown in Table I. The learning curves of models A, B, and C are depicted in figures 4, 5,

and 6, respectively. It can be seen that model A started to overfit the training data after almost 40 epochs. Fig. 4 shows that after 40 epochs the validation loss begins to increase whereas the training loss keeps on declining, implying that overfitting is happening. In model B however, this phenomenon is pushed away by almost 20 epochs. As it can be seen on Fig. 5, the validation loss keeps on declining with the training loss showing that the data augmentation layers used in this model are properly regularizing the model and avoiding the overfitting in the first 60 epochs. Model C closely follows model B since the only difference between the two models is that model C incorporates an early stopping method to check and stop the training should overfitting happen. Since the data augmentation method used in model B is working well, overfitting will not happen until 50 epochs of this model but training stops after 50 epochs to avoid overfitting. There are no significant differences between model B and C in the first 50 epochs of training. Comparing Figures 5 and 6 confirms that these methods are both trained in the same way. Fig. 7 shows a sample of the test data and the generated masks using the three discussed models.

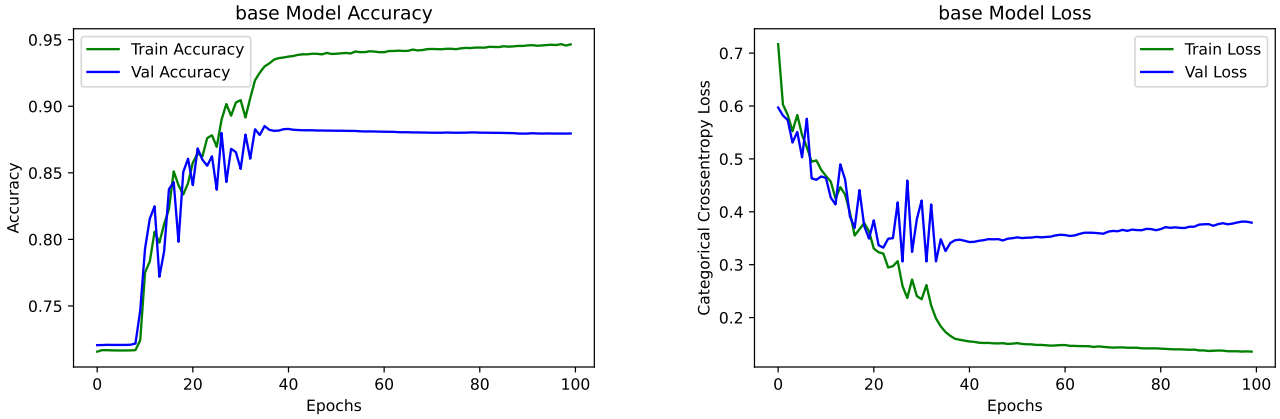

Fig. 4. Learning curves for model A.

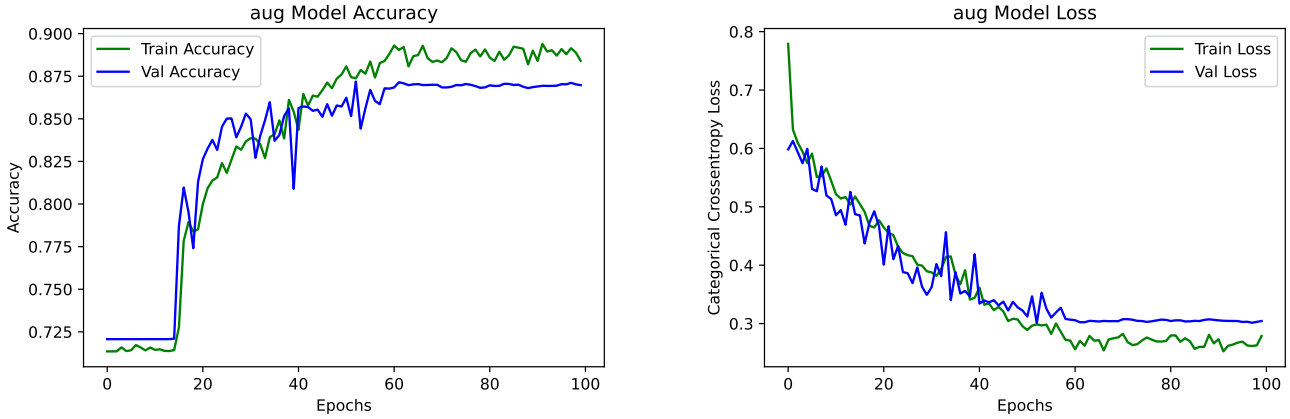

Fig. 5. Learning curves for model B.

IV. Conclusion

This work experimented with three different models to segment humans in images, analyze the overfitting issue in the models, and how it can be overcome by using regularization methods such as data augmentation and early stopping. The experimental results show that by utilizing synthetic data, the network can learn a generalizable pattern in the image data and reduce the chance of overfitting. To further improve this work, it is possible to analyze how other regularization methods such as dropout, L1, and L2 regularization can affect the performance and generalizability of the model.

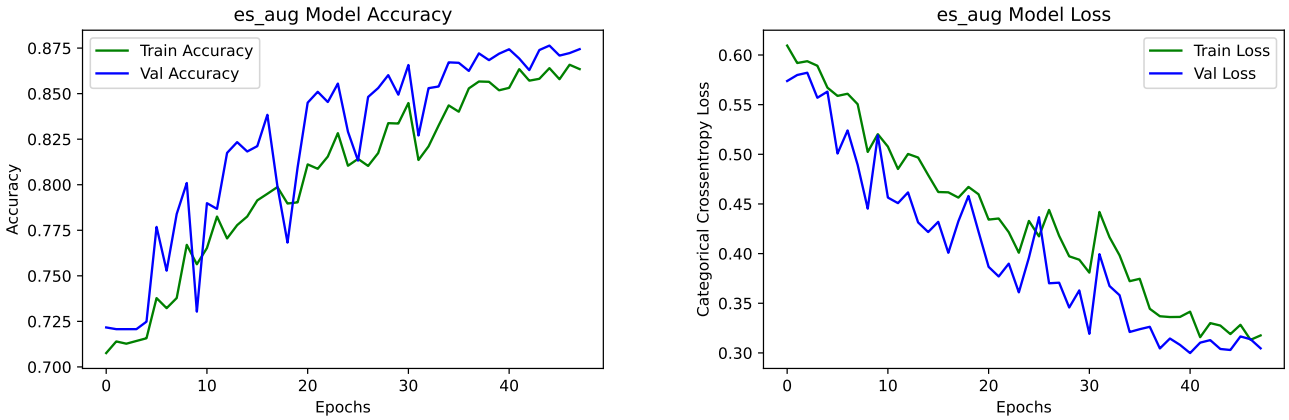

Fig. 6. Learning curves for model C.

Fig. 7. Sample test data and model A, B, and C predictions from top to bottom.

REFERENCES

[1] V. Dumoulin and F. Visin, "A guide to convolution arithmetic for deep learning," *arXiv preprint arXiv:1603.07285*, 2016.

[2] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[3] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.