U-Net for Satellite Image Segmentation: Improving the Weather Forecasting

Yue Zhao
Department of Electrical and Computer Engineering
University of Rochester
Rochester, US
yzhao88@ur.rochester.edu

Zhongkai Shangguan
Department of Electrical and Computer Engineering
University of Rochester
Rochester, US
zshangg2@ur.rochester.edu

Wei Fan
Department of Electrical Engineering
Columbia University
New York, US
wf2271@columbia.edu

Zhehan Cao
Department of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, US
zcao76@gatech.edu

Jingwen Wang
Department of Electrical and Computer Engineering
University of Rochester
Rochester, US
Jwang191@ur.rochester.edu

2020 5th International Conference on Universal Village (UV) | 978-1-7281-9523-0/20/$31.00 ©2020 IEEE | DOI: 10.1109/UV50937.2020.9426212

***Abstract*—The clouds organization plays a huge role in forecasting the weather and Earth's future climate; therefore developing a better intelligent model is a way to accurately predict weather and predict weather and meteorological disasters, such as hurricane and tornado. In this paper, we classified the patterns of clouds into four types (sugar, flower, fish, and gravel) proposed by Rasp et al. and performed image segmentation. All the datasets were adopted from the Kaggle Competition. U-net was used as the basic structure and ResNet was applied to the original U-net structure after the data analysis. In addition, three different loss functions were used for training, the Test-time Augmentation was performed before feeding the test data to the model and the Amendment method was used to modify the results. The final dice coefficient reaches up to 0.665, which is an outstanding outcome that reflects the robustness of our method and training.**

Keywords*—*clouds organization, image segmentation, ResNet, U-net, loss functions, Test-time Augmentation, amendment

I. Introduction

Cloud is one of the major weather forecast factors that people concern for thousand of years. With growing technology in recent years, there are many techniques to analyze different cloud patterns from satellite images. With the main principle of the Universal Village, developing intelligent models for satellite image segmentation better connects humans with the environment. For example, meteorologists analyze the satellite images and produce public warnings on imminent disasters, such as a tornado, typhoon, flooding and snowstorm, subsequently minimizing the economic loss associated with these extreme weather events. The satellite image segmentation typically employs three techniques: active contours, threshold technique, and K-means technique. However, the accuracy and stability of these three techniques are limited [1]. Therefore, we are developing a new method to do the satellite image segmentation. U-net was originally from biological imaging segmentation with better accuracy, which inspired us to apply U-net in the satellite image segmentation.

In this paper, we modify the U-net structure by involving Resnet34 in the down-sampling part. As the clouds' organization plays a critical role in determining the Earth's climate, the subjective pattern classification is essential for weather prediction and analysis. The four subjective patterns of organization in this experiment were defined as sugar, flower, fish, and gravel as Fig. 1 shows [2]. All the datasets used in the training were from the Kaggle competition [3]. Three different loss functions were used for training, the Test-time Augmentation was performed before feeding the test data to the model and the Amendment method was used to modify the results. The final dice coefficient reaches up to 0.665.

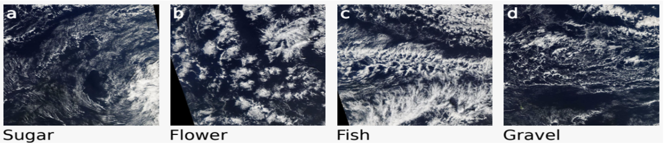


Fig. 1. Four cloud organization patterns

This paper is organized as follows. Section II introduces the methods of data analysis. Section III introduces the models and methods used in the experiment. Section IV describes the training methods for both the segmentation and classification. Section V presents the testing results of the experiment, followed by the discussion and conclusion in Section VI.

II. Models and Methods

Fully Convolutional Networks (FCNs) by themselves, trained end-to-end, pixels-to-pixels, exceed the state-of-the-art in semantic segmentation [4].

A. U-net

U-net keeps the main structure of FCNs, which supplement a usual contracting network by successive layers, where the

978-1-7281-9523-0/20/$31.00 ©2020 IEEE

decoder part is replaced by upsampling operators [5]. The advantage of U-net is that it yields more precise segmentation with fewer training images processed. To be specific, U-net combines the location information from the downsampling path with contextual information in the upsampling part, so that the context information is prorogated to the higher resolution layers. As illustrated in Fig. 2, the architecture of U-net is almost symmetric which yields a u-shaped structure. Moreover, U-net uses excessive data augmentation by applying elastic deformations to the available training images [6]. This process allows the network to learn invariance to such deformations with no need to observe the transformations in the annotated image corpus.

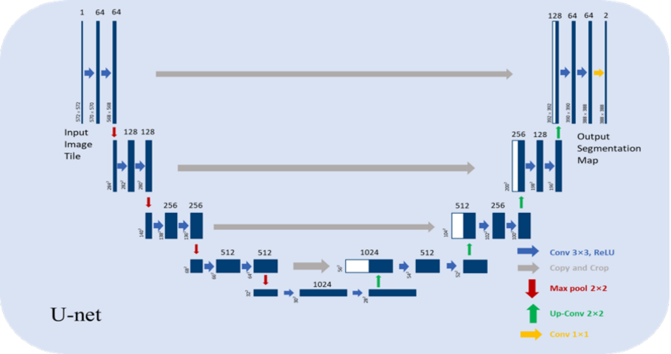


Fig. 2. Illustration of U-net structure

B. ResNet34

Based on the data analysis, the 34-layer residual nets (ResNet34) is chosen as the encoder part of the entire U-net network in the experiment. ResNet34 is a residual network with 34 parameter layers, which exhibits considerably low training [7]. Fig. 9 shows that the ResNet34 consists of one convolution and pooling step followed by four layers of similar behavior. Each layer contains a different number of the residual block since it normally increases the number of convolutions within a block when ResNet get deeper. Note that the number of the total layers remains as four times as Fig. 3 shows.

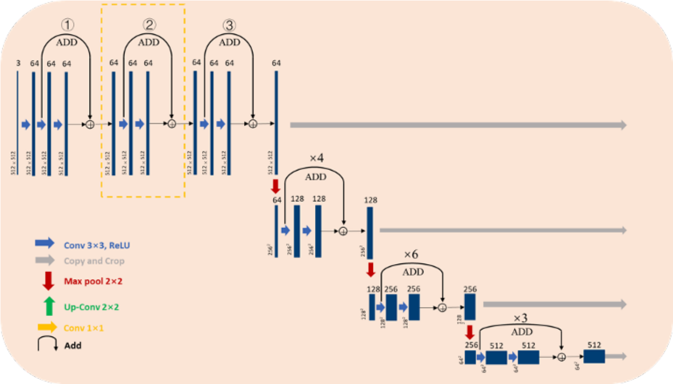


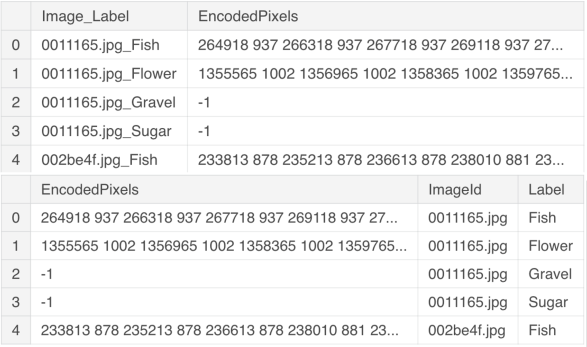
Fig. 3. Illustration of ResNet structure

The main difference between our ResNet34 and the plain network of 34 layers is that there are two 3 × 3 convolutions and ReLU activation in each residual block and the output of each block is the input image added to the convolved one. Therefore, the input of the next residual block is the output from the last block, and the dimensions of width and height remain constant in the entire layer.

III. Data Analysis

This section was divided into three parts, which includes run-length encode, data balance analysis and type analysis.

Firstly, run-length encode was defined as a data compression algorithm that is supported by most bitmap file formats. The reason for using run-length encoding is that the datasets used in the implement were got from Kaggle competition website [3]. The original datasets obtained from the website are as the Fig. 4 shown below, which provides the image labels and the encoded pixels only. As the Fig. 4 shown that the information provided from the original datasets were all divided and organized into three categories, which are Encoded Pixels, image ID and label.



	Image_Label	EncodedPixels
0	0011165.jpg_Fish	264918 937 266318 937 267718 937 269118 937 27...
1	0011165.jpg_Flower	1355565 1002 1356965 1002 1358365 1002 1359765...
2	0011165.jpg_Gravel	-1
3	0011165.jpg_Sugar	-1
4	002be4f.jpg_Fish	233813 878 235213 878 236613 878 238010 881 23...

	EncodedPixels	ImageId	Label
0	264918 937 266318 937 267718 937 269118 937 27...	0011165.jpg	Fish
1	1355565 1002 1356965 1002 1358365 1002 1359765...	0011165.jpg	Flower
2	-1	0011165.jpg	Gravel
3	-1	0011165.jpg	Sugar
4	233813 878 235213 878 236613 878 238010 881 23...	002be4f.jpg	Fish

Fig. 4. Original dataset for one image

For the encode pixels shown in Fig. 4, the -1 value stands for this image did not include this pattern. The following steps were taken in order to decompress the encode pixels array information from Fig. 5 to a binary mask: firstly the index range was needed to be set as number of images, then the all zero image matrix was also set with respect to the shape of the original image, and then run length code was able to tell us the locations for 1. For example, the first number of encoded pixels array (shown in Fig. 4) is the starting pixel for 1, then the second number of the array is the length of 1 (count the pixel horizontally), and then the third number of the array is another starting pixel for 1, and the fourth number is the length of 1, and keep doing the same work for all the remaining encoded pixels number in the array. After that, all the other positions were stay as 0, therefore, the mask came out as the image shown in Fig. 5.

Secondly, aiming to avoid that the distribution was heavily polarized to only one side, which is not helpful for machine learning, data balance analysis was necessary to be performed. As the Fig. 6 shown below, the expected data distribution in the implement was evenly distributed that contained 23.5% of fish pattern, 19.98% of flower pattern, 24.83% of graver pattern and 31.69% of sugar pattern. In addition, doing the noise analysis and checking the validation of the image is essential such as some noise were shown in the middle of Fig. 7, therefore it was decided that the images with noise larger than 50% were abandoned from the dataset.

Fig. 5. Mask image outputed

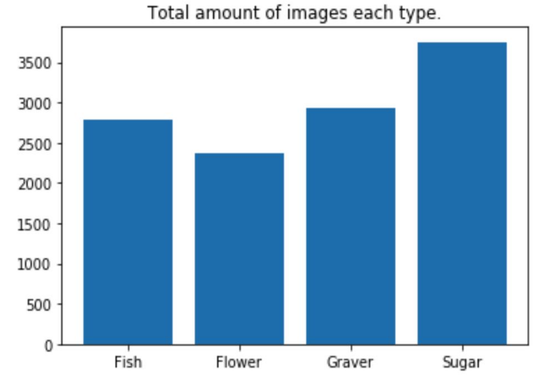


Fig. 6. Data balance analysis

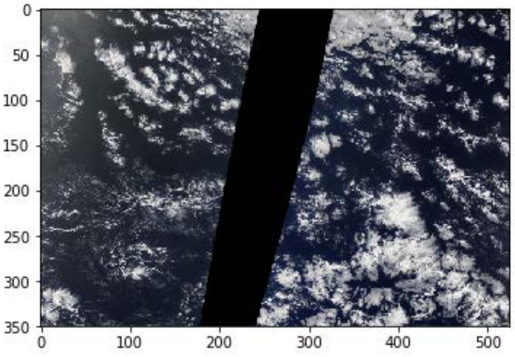


Fig. 7. Noise analysis

Moreover, the type analysis needed to be taken as each image and each pixel can have multi-types. To be specific in multi-type images, in Fig. 8, the first bar shows the quantity of images that only contain one type of pattern, the second bar shows the quantity of images that contain two types of pattern, the third bar shows the quantity of images that contain three types of pattern, and the fourth bar shows the quantity of images that contain four types of pattern. As the four patterns have some overlaps with established modes of organization, the Fig. 9 shows that each pixel could have multi-types that the first bar shows the quantity of images that have no overlap, the second bar shows the quantity of images that have one overlap, the third bar shows the quantity of images that have two overlaps while the fourth bar shows the quantity of images that have three overlaps. In the case that the images don't have overlap, Softmax was used as the classification method to implement. The Softmax has five independent types, where the four types were indicated before and the background was used as the fifth type. In the case that overlap involved, Sigmoid was used as the classification method to implement and it doesn't need for type background as the input.

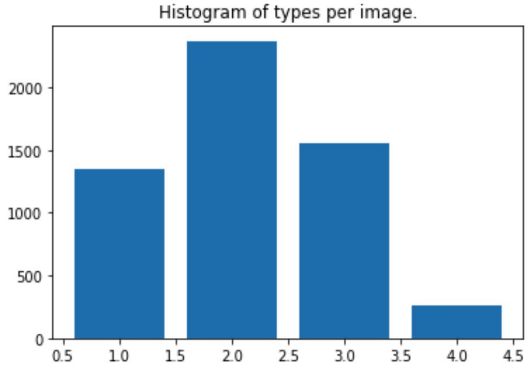


Fig. 8. Histrogram of types per image

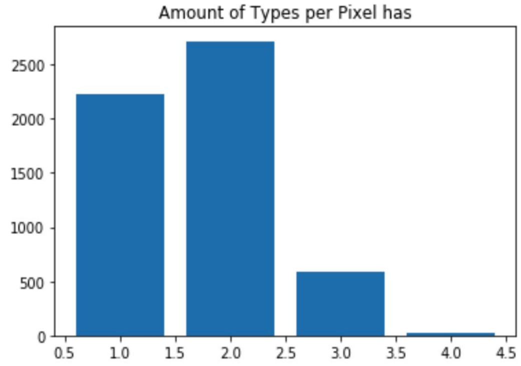


Fig. 9. Amount of types per pixel

IV. Training Methods

Training strategy plays an important role in machine learning. A good performance of a model is inseparable from well training methods, even with advanced model architectures.

Training a neural network optimizes values for the weights and bias from labeled data in order to map a particular input to some output. The training of a neural network model requires four specifications: input data source with the label need to be predicted, loss function, optimizer, and learning rate scheduler.

A. Data Augmentation

Recent advances in deep learning models have been largely attributed to the quantity and diversity of data gathered in recent years [9]. Image augmentation increases the quantity and improves the diversity of data available for the training models. Such feature is especially effective when dealing with the dataset with limited images.

Without collecting new data, some common augmentation techniques including rotation and flip are applied. Based on data analysis, we discover that the shape and density of the clouds should contain the most essential features. Therefore, we use contrast change techniques to distinguish clouds from the background, i.e. sky, including histogram equalization and contrast limit adaptive histogram equalization (CLAHE).

B. Evaluation Metrics

We evaluate the performance of our model by using the mean of the Dice coefficient, which is widely used to compare the pixel-wise agreement between a predicted segmentation and its corresponding ground truth. The mathematical equation is defined by:

$$Dice\ Coefficient = \frac{2 \times |X \cap Y|}{|X| + |Y|}, \tag{1}$$

where X is the predicted set of pixels and Y is the ground truth.

C. Loss Functions

Neural networks are trained using gradient descent based backpropagation and require a loss function when taking algorithm from theoretical to practical. The loss function reflects the distance between the predicted results and corresponding truth, and is usually defined according to the evaluation metrics. We perform a combination of Binary Cross-Entropy (BCE) and soft Dice loss as we are solving a multiclass segmentation problem. The total loss is defined by:

$$\begin{aligned} Total\ loss = & -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot \log\left[p\left(y_i\right)\right] \\ & + \left(1 - y_i\right) \cdot \log\left[1 - p\left(y_i\right)\right] \\ & + 1 - \frac{2 \cdot \sum_i^N y_i \cdot p\left(y_i\right)}{\sum_i^N y_i^2 + \sum_i^N p\left(y_i\right)^2} \end{aligned} \tag{2}$$

where N is the number of pixels in one image, y_i and $p(y_i)$ is the ground truth and predicted values for i-th pixel, respectively. We assign the BCE and soft Dice loss the same weight.

D. Optimizer and Learning Rate Scheduler

We perform Adam as the optimizer. Adam is a very popular algorithm in the field of deep learning because it can achieve an excellent results quickly. Empirical results prove that the Adam algorithm has excellent performance in practice and has advantages over other types of random optimization algorithms [9].

The Learning rate is a hyper-parameter that controls how much we are adjusting the weights of our network with respect to the loss gradient. The model can learn faster but at a sacrifice of accuracy. on the contrary, a slow learning rate usually achieves better performance but is time-consuming. In this way, we adjust the learning rate each iteration by using warm restart, which is also known as cyclical learning rates, the mathematical equation is given by [10]:

$$\eta_t = 0.5 + 0.5 \cos\left(\frac{T_{cur}}{T_i}\pi\right), \tag{3}$$

where T_{cur} indicates how many epochs passed since the last restart, and T_i is the epoch of the next restart.

E. Ensemble Learning Strategy

Ensemble learning is the process to combine multiple models in order to solve a computational intelligence problem. Ensemble learning is widely used to improve the performance of a model. We perform Test Time Augmentation (TTA) and stacking techniques.

TTA is an application of data augmentation to the test dataset which involves creating multiple augmented copies of each image in the test set, predicting the model, and returning the average of those predictions [8]. To be specific, horizontal flipping, vertical flipping, blur were used during our result resembling process.

Stacking involves training a learning algorithm to combine the predictions of several other learning algorithms [11]. Stacking typically yields performance better than any single one of the trained models. Specifically, we use the grid search algorithm to determine the best threshold for the prediction values from the segmentation model as they are given by probability. Besides, we train another classification model to fix the segmentation model errors.

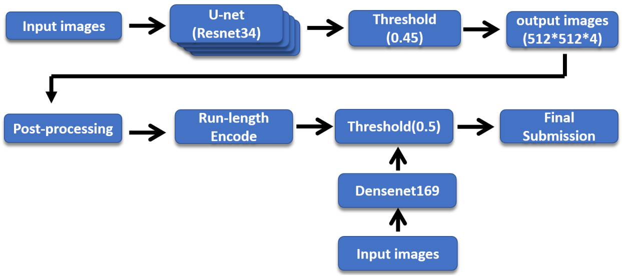


Fig. 10. Inference and post-processing flowchart

The overall inference process including TTA and ensemble learning is shown in Fig. 4. Fig. 4 shows that the classification model was trained based on Densenet169 with loss function Binary Cross-Entropy and metric Area Under the Curve (AUC). This classification model gives the probability that represents whether each image would contain a certain class. The result would be kept if the probability is greater or equal than 0.5, and the encoded pixels were manually deleted if the probability if less than 0.5.

V. Results

For this neural network, the input data are the original image data as well as their labels. However, the dataset obtained is limited in the quantity of input images (5546 original images); therefore, data augmentation was used to solve this problem. Flipping, rotation, and contrast change techniques were chosen in our experiment as discussed in Section III, A. We randomly choose different images to do the data augmentation in each epoch such as 50% flipping, 50% rotating, and 10% random CLAHE. Therefore, there are different input images generated for each epoch, which largely enhances the robustness of the training process. For the neutral network input labels, the four types of masks generated from run-length encode data individually and then concatenated into a $512 \times 512 \times 4$ label.

For the application of ResNet34, the input of the entire structure is the RGB image ($512 \times 512 \times 3$). After padding and 2×2 max pooling, the image was convolved by a 7×7 matrix, and finally generate the standard data size $512 \times 512 \times 64$ images. Therefore, the output from each layer is then passed through a 2×2 convolution operation. After the four times repeated works described above, the output of $16 \times 16 \times 512$ was generated.

The modification of the U-net structure in the up-sampling part was kept since this structure enables a more precise output of the image. The structure has many feature channels, which allows the network to propagate the context information to the higher resolution layers, and this strategy allows the seamless segmentation of arbitrarily large images by an overlap-tile strategy [5].

Fig. 11 presents one results from our experiment. In this example, the image contains 3 types of clouds. The left column images show the ground truth of cloud patterns (shadowed by white), which are fish, flower, and sugar. The four images on the right demonstrate the prediction of our model. It is perceived that the high extent of alignment between the ground truth and the predictions indicates the strength of our model.

The dice coefficient of our model reaches 0.665 which shows this model can distinguish different types of cloud patterns efficiently and accurately.

VI. Conclusion and Discussion

This paper presents the image segmentation for classifying four types of clouds patterns (sugar, flower, fish, and gravel). U-net was used as our basic structure and ResNet34 is the structure applied additionally to the original structure after the data analysis. All the datasets were provided by Kaggle Competition [3] and three different loss functions (Binary Cross-Entropy, Dice Loss, and Jaccard Loss) were used for training. Additionally, the Test-time Augmentation was performed before feeding the test data to the model and the Amendment method was used to modify the results. The final dice coefficient reaches up to 0.665, which is an outstanding outcome helps us ensure that our experiment performed well in method and training.

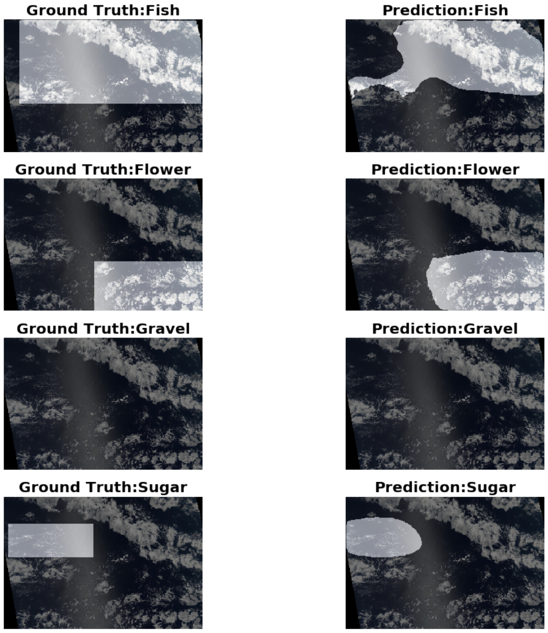


Fig. 11. Sample testing results from two original images

There are two limitations in this experiment. One is that the gravel and sugar patterns were similar to each other, which makes the distinguishing process becomes more complicated. The second limitation is that the program we developed demands a high standard for the original datasets because our experiment requires the amount of images in datasets distributed evenly in each type of cloud pattern.

References

[1] N. P. Deepika and K. Vishnu, “Different techniques for satellite image segmentation,” 2015 Online International Conference on Green Engineering and Technologies (IC-GET), Coimbatore, 2015, pp. 1-6, doi: 10.1109/GET.2015.7453836.

[2] S. Rasp, H. Schulz, S. Bony, and B. Stevens, 2020. “Combining crowd-sourcing and deep learning to explore the meso-scale organization of shallow convection,” Bulletin of the American Meteorological Society, preprint 2020.

[3] “Understanding clouds from Satellite Images,” online Available at: https://www.kaggle.com/c/understanding/cloud/organization.

[4] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, 2015, pp. 3431-3440, doi: 10.1109/CVPR.2015.7298965.

[5] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation,” International Conference on Medical Image Computing Computer-assisted Intervention 2015.

[6] A. Dosovitskiy, P. Fischer, J. T. Springenberg, M. Riedmiller, and T. Brox, “Discriminative Unsupervised Feature Learning with Exemplar Convolutional Neural Networks,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 38, no. 9, pp. 1734-1747, 1 Sept. 2016, doi: 10.1109/TPAMI.2015.2496141.

[7] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90.

[8] J. Brownlee, “How To Use Test-Time Augmentation To Make Better Predictions,” Machine Learning Mastery, Available at https://machinelearningmastery.com/how-to-use-test-time-augmentation-to-improve-model-performance-for-image-classification/.

[9] D. Seita, “1000X Faster Data Augmentation,” The Berkeley Artificial Intelligence Research Blog, Available at https://bair.berkeley.edu/blog/2019/06/07.

[10] L. N. Smith, “Cyclical Learning Rates for Training Neural Networks,” 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), Santa Rosa, CA, 2017, pp. 464-472, doi: 10.1109/WACV.2017.58.

[11] R. Polikar, “Ensemble Learning,” scholarpedia, Available at http://www.scholarpedia.org/article/Ensemble_learning.