

ROBUST WIND TURBINE BLADE SEGMENTATION FROM RGB IMAGES IN THE WILD

Raül Pérez-Gonzalo^{1,2}, Andreas Espersen², Antonio Agudo¹

¹Institut de Robòtica i Informàtica Industrial, CSIC-UPC, Spain

²Wind Power LAB, Copenhagen, Denmark

ABSTRACT

With the relentless growth of the wind industry, there is an imperious need to design automatic data-driven solutions for wind turbine maintenance. As structural health monitoring mainly relies on visual inspections, the first stage in any automatic solution is to identify the blade region on the image. Thus, we propose a novel segmentation algorithm that strengthens the U-Net results by a tailored loss, which pools the focal loss with a contiguity regularization term. To attain top performing results, a set of additional steps are proposed to ensure a reliable, generic, robust and efficient algorithm. First, we leverage our prior knowledge on the images by filling the holes enclosed by temporarily-classified blade pixels and by the image boundaries. Subsequently, the mislead classified pixels are successfully amended by training an on-the-fly random forest. Our algorithm demonstrates its effectiveness reaching a non-trivial 97.39% of accuracy.

Index Terms— Blade Segmentation, Wind Turbine, Blade Inspections, BU-Net, Hole Filling.

1. INTRODUCTION

Wind energy has demonstrated to be an excellent alternative energy source, thanks to being completely renewable and environmentally sustainable [1]. The increase in installed capacity urges to adapt to more demanding challenges in terms of cost and logistics [2], with special focus on developing non-destructive inspection manners for health monitoring. Drone inspections have become a fast and reliable setup for wind turbine diagnosis [3], capturing around 400 high-resolution images per turbine to design a repair plan. All in all results in high-volume data that encourages the development of automated, scalable and cost-effective solutions.

To successfully design a data-driven maintenance framework for wind turbines in operation, current researchers are tackling distinct image-based problems such as blade defect detection [4, 5] or autonomous navigation [6, 7]. However, current solutions have several limitations and are not being applied yet to the industry. Hence, these challenges must be simplified by means of distinct low-level vision techniques.



Fig. 1. Wind turbine blade images in the wild with distinct complexity due to the variation of the landscapes, the blade shape and size, and the presence of shadows or dirt.

To this aim, this paper proposes addressing segmentation to reduce the complexity of these learning tasks.

Current state-of-the-art segmentation methods rely on learning-based approaches, usually employing an encoder-decoder architecture [8, 9, 10] and, recently, an attention mechanism on the encoder [11, 12]. With a properly tailored architecture, these models have proven to successfully obtain high precision results. Unfortunately, they do not provide a segmentation algorithm that generalizes for newly acquired images [13]. A generic solution is essential to deal with a large variety of images (see Fig. 1) obtained on regular wind turbine inspections under a different setup.

A common technique to enhance the segmentation mapping is including a post-processing step. DeepLab [8] utilized Conditional Random Fields to fine-tune the localized segmented objects. DeepLab V3+ [14] is further strengthened through superpixel merging in [13]. Another interesting approach is ResNeSt [15], which incorporates a channel-wise attention with multi-path representation in its last stage. However, these methods do not leverage our prior information about blade images, such as that the blade traverses the image or that images of the same inspection share image properties.

This work proposes a novel segmentation algorithm that identifies the blade region in very diverse images in appearance. First, a preliminary mask is obtained by a customized U-Net model [16], whose architecture still inspires current techniques [17]. Then, this preliminary result is improved through a set of post-processing steps: an extended hole filling routine from [18] followed by an unsupervised random forest [19]; refined again through another hole filling step. The hole filling relies on properly identifying the blade image borders and the random forest on underfitting the preliminary masks from the same blade surface. As a result, we obtain an accurate, robust and efficient model for blade segmentation that can handle a wide variety of images in the wild.

This work has been supported by the Innovation Fund Denmark under 2021 ID1044-0044A and by the project MoHuCo PID2020-120049RB-I00 funded by MCIN/AEI/10.13039/501100011033.

2. WIND TURBINE BLADE SEGMENTATION

Let \mathbf{I} be a $H \times W \times 3$ color image of a wind turbine blade that was captured from an arbitrary point of view and under uncontrolled lighting conditions. Our goal is to recover a $H \times W$ binary mask $\hat{\mathbf{S}}$ to indicate the region in the image that belongs to the blade. In our formulation, the mask is represented by the binary entries $\hat{s}_{h,w}$, where $h = \{1, \dots, H\}$ and $w = \{1, \dots, W\}$ denote the pixel coordinates in the mask.

Given the corresponding ground-truth mask \mathbf{S} , we propose a supervised learning framework composed of four modules to learn the mapping $\mathbf{I}_{i,j} \rightarrow \mathbf{S}_{i,j}$ for an i -th blade and j -th image: 1) an encoder-decoder network algorithm with a tailored loss, 2) a hole filling step, 3) a random forest block and, 4) a latter hole filling step to refine the solution. Without loss of generality, for every blade surface there are considered up to J images for the i -th blade.

2.1. Blade U-Net (BU-Net)

Given an input RGB image \mathbf{I} , our BU-Net aims at classifying each input pixel into blade or background, i.e., obtaining the output mask $\hat{\mathbf{S}}^{BU}$ by exploiting the labels in \mathbf{S} . This module follows a standard U-Net architecture [16], and its last layer includes a 1×1 convolution to map the feature channels to the desired number of classes. The mask $\hat{\mathbf{S}}^{BU}$ is obtained by combining four predictions via soft voting, where each one corresponds to a flipping rotation. Additionally, a regularized focal loss is employed to adapt the learning task to our particular dataset. The two terms are detailed below.

The *Focal Loss* optimizes our segmentation mapping [20] by penalizing low-confidence predictions. Moreover, to avoid benefitting the majority class (in our case, the background) relative to the minority class, we apply weights to each class:

$$\mathcal{L}_f(\mathbf{S}, \hat{\mathbf{S}}^{BU}) = - \sum_{h,w} \left[\alpha (1 - \sigma(\hat{s}_{h,w}^{BU}))^\gamma s_{h,w} \log \sigma(\hat{s}_{h,w}^{BU}) + (1 - \alpha) (\sigma(\hat{s}_{h,w}^{BU}))^\gamma (1 - s_{h,w}) \log(1 - \sigma(\hat{s}_{h,w}^{BU})) \right], \quad (1)$$

where $\sigma(\cdot)$ denotes the sigmoid function, γ controls the confidence penalty; and α sets the class balance.

The *Contiguity Loss* [21] minimizes the jaggies in the blade boundaries, forcing to detect contiguous objects:

$$\mathcal{L}_c(\hat{\mathbf{S}}^{BU}) = \frac{1}{HW} \left[\sum_{h,w} (\sigma(\hat{s}_{h+1,w}^{BU}) - \sigma(\hat{s}_{h,w}^{BU}))^2 + \sum_{h,w} (\sigma(\hat{s}_{h,w+1}^{BU}) - \sigma(\hat{s}_{h,w}^{BU}))^2 \right]^{1/2}. \quad (2)$$

The *Total Loss* is defined as a linear combination of the two previous losses modulated by the scalar λ :

$$\mathcal{L} = \mathcal{L}_f + \lambda \mathcal{L}_c. \quad (3)$$

As the output of this block is the probability for the blade class, i.e., $\sigma(\hat{\mathbf{S}}^{BU})$, it is quantized by a threshold of 0.4.

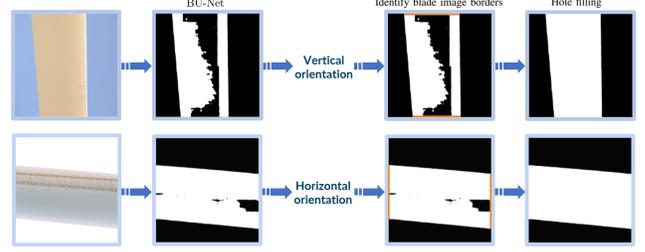


Fig. 2. Hole filling representation. First, the blade orientation is detected to identify the blade image borders (in orange). Then, holes that edge with the image borders are filled.

2.2. Hole Filling

The estimated segmentation $\hat{\mathbf{S}}^{BU}$ still contains some non-realistic regions, including some holes. We propose a *hole filling* step to reduce these artifacts. Our approach is able to not only fill in the holes surrounded by blade pixels, but also when they are surrounded by the image borders. By exploiting a spatial prior, our approach relies on localizing first the blade pixels in the image boundaries, i.e., the blade-borders.

Assuming that the blade traverses the image, we first compute the blade orientation by means of an accumulated gradient. Basically, it is classified as vertical (horizontal) if the accumulated gradient along the x -axis is higher (lower) than along the y -axis. Once the orientation is identified, we know the blade-border locations, so we can enforce their continuity by setting their pixels as blade. Finally, we proceed with the standard hole filling algorithm [18] to obtain the output $\hat{\mathbf{S}}^H$. Both vertical and horizontal orientation examples are displayed in Fig. 2. The hole filling step is applied twice: 1) considering as input the estimation in our BU-Net and, 2) using as input the output of our random forest algorithm that is introduced below. To differ between both hole filling estimations, we will use the notation $\hat{\mathbf{S}}^{H1}$ and $\hat{\mathbf{S}}^{H2}$, respectively.

2.3. Unsupervised Random Forest

Finally, we propose a simple random forest to denoise the estimation and to fix uncommon errors of a single image. The inability of previous steps to capture global context is tackled by learning which RGB pixels are usually mapped as foreground, providing a fine-grained boundary precision. To this end, we learn the mapping $\{\mathbf{I}_{i,j}\}_{j \in J} \rightarrow \{\hat{\mathbf{S}}_{i,j}^{H1}\}_{j \in J}$ given the i -th blade surface. This model ensembles tree estimators by averaging their probabilistic predictions to set whether a single RGB pixel and its local neighborhood is foreground:

$$P^{RF}(\hat{s}_{h,w}^{RF}) = \frac{1}{|RF|} \sum_{t \in RF} P^t(\hat{s}_{h,w}^t | \hat{s}_{h,w}^{H1}, \text{neighbors}(\hat{s}_{h,w}^{H1})),$$

where RF is the set of decision trees t . It is worth pointing out that this approach enables our algorithm to adapt to the image features of a specific i -th blade. Once the model is trained, it is applied to obtain the estimation $\hat{\mathbf{S}}_{i,j}^{RF}$.

Lastly, we again apply a hole filling (section 2.2) step to refine the solution, obtaining $\hat{\mathbf{S}}^{H2}$ as our final estimation $\hat{\mathbf{S}}$.

3. EXPERIMENTAL RESULTS

In this section, we show the effectiveness of our segmentation approach through both quantitative and qualitative evaluations as well as showing a comparison with competing techniques, providing the results for every step in our algorithm.

3.1. Dataset

We present a dataset that consists of high-resolution images taken by drone and ground-based equipment from distinct locations of the blade. The images provide a large variety of blade sizes, shapes, illumination conditions, as well as the background to be observed (Fig. 1). Another background variation arises from on-shore and off-shore wind turbines. For this challenging dataset, we propose 1,712 images for training, 120 for validation and 200 for test. As a segmentation ground-truth is not provided, we manually annotated all the images. To make a fair analysis, the three sets are composed of different windfarms and inspection campaigns, ensuring their independence and that the test properly generalizes. In particular, we randomly selected 20 images of each windfarm to generate the validation and test sets. On the other hand, the training data was selected from a pool of different blade images, prioritizing the ones that are more challenging. These are the images in the root, the tip or the max-cord (where the blade section is widest). All color images and masks were min-max normalized. In addition to that, flipping and cropping strategies were applied to learn the desired invariance and robustness properties.

3.2. Implementation Details

The original input images were resized to 1024×1024 pixels. We use Adam solver [22] with an initial learning rate of 10^{-4} and a mini-batch of 1. A custom scheduler is employed to reduce the learning rate when there is no improvement on the validation loss after three epochs. Early stopping is performed. The model is trained using a NVIDIA GeForce RTX 3080 GPU. The weights for the focal loss in Eq. (2.1) are set to $\gamma = 2$ and $\alpha = 0.25$, and the weight in Eq. (3) $\lambda = 1$.

3.3. Quantitative Evaluation

The segmentation performance per image is analyzed in terms of accuracy, recall and F1-score.

Hyperparameter Search. Our random-forest step (section 2.3) is optimized in terms of the number of split branches and the number of decision trees, obtaining the highest accuracy with five tree estimators that have a maximum of four split branches. Additionally, when the local pixel neighbors are included in the input, we enhance the random forest masks. However, increasing the number of input neighbors does not increase the segmentation accuracy. Thus, we took as input the local pixel and its most immediate neighbors.

Test Set Results. The performance over the test set is compared between each algorithm step. In particular, we report the accuracy, recall and F1-score in Table 1.

Table 1. Test results after each segmentation step.

Performance metric	BU-Net \hat{S}^{BU}	Fill holes \hat{S}^{H1}	Random forest \hat{S}^{RF}	Fill holes 2 \hat{S}^{H2}
Accuracy (%)	92.28	96.11	97.11	97.39
Recall (%)	85.73	92.08	92.37	93.35
F1-score (%)	90.41	94.53	95.12	95.73
mIoU (%)	87.71	93.57	95.07	95.53

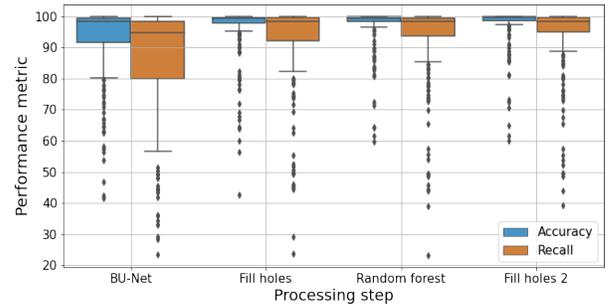


Fig. 3. Box plot test results after each segmentation step.

Table 2. Quantitative comparison on blade segmentation. Relative metrics are with respect to the worst estimation.

Method	Accuracy %	Recall %	F1-score %	mIoU %	Relative accuracy	Relative recall
U-Net [16]	86.24	68.93	77.95	79.94	1	1
DeepLabv3+ [14]	94.14	87.38	89.03	90.74	1.09	1.27
SW [9]	93.48	91.71	91.37	89.71	1.08	1.33
ResNeSt [15]	94.23	91.47	92.77	91.20	1.09	1.33
U-NetFormer [17]	96.20	93.51	94.42	93.65	1.12	1.36
Ours \hat{S}	97.39	93.35	95.73	95.53	1.13	1.35

By only employing our BU-Net, we obtain a notable outcome of 92.28% of accuracy. However, this result is not sufficiently good for blade assessments, because the recall of 85.73% could compromise the detection of defects. For instance, if a large area of the blade is masked as background, it could cover the region where there are structural damages, which is a severe defect that cannot be overlooked.

By incorporating the first hole-filling step, we substantially improve performance. Namely, the recall increases to 92.08%, which represents the highest gain of all the post-processing steps. This huge difference is due to the BU-Net being able to successfully capture the blade borders, but not the inner region of the blade. Further algorithm steps are necessary to yield further improvement, reaching in the end a novel performance of 97.39% of accuracy and 93.35% of recall. On balance, and as it can be seen in Table 1, every step is key to sort out the problem properly, obtaining a final accurate solution that cannot be achieved directly by our BU-Net.

Fig. 3 analyzes the performance dispersion of each step. We observe the skewness of the BU-Net performance: rather spread along the tail with lower performance, containing many instances with poorer accuracy and recall compared with their median. As we include the different steps in our algorithm, that dispersion is reduced notably. Therefore, despite not substantially improving the mean performance, the subsequent steps are the cornerstone to ensure robustness.

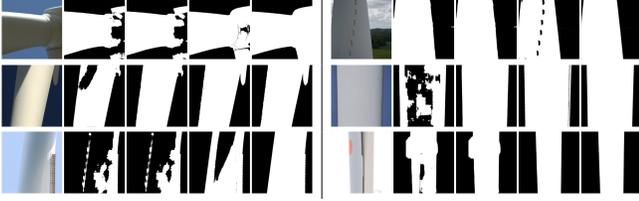


Fig. 4. Qualitative evaluation on test images after applying each step of our proposed segmentation algorithm. On both sides, the same information is displayed. **First column:** Input color image (I). **From second to fifth column:** BU-Net (\hat{S}^{BU}), first hole filling (\hat{S}^{H1}), random forest (\hat{S}^{RF}) and second hole filling ($\hat{S}^{H2} \equiv \hat{S}$) estimations, respectively.

Comparative Results. As the BU-Net employs the U-Net architecture [16], we compared our proposed algorithm with the original one and state-of-the-art methods (Table 2). We outperform those methods, thanks to overcoming the blade-background imbalances, and escaping from the tendency of predicting the most common region.

3.4. Qualitative Results

We visualize the segmentation results of each algorithm step in Fig. 4. In general, our BU-Net can capture the blade edges and, therefore, its overall shape. However, it struggles with the inner area of the blade, which is addressed in the first hole filling step. Note that just using the standard hole filling approach would not improve the BU-Net outputs. Then, the random forest is responsible for fixing the misplaced pixels, acting as a regularizer of the pixel color intensity. By training a simple model, we ensure there is no overfitting and that it just learns the main patterns. Hence, if the BU-Net can in general classify which color intensities belong to the blade region, then the random forest would learn those patterns and resolve the particular images or pixels that failed, obtaining robustness in the solution. The second hole filling step is applied to solve the shortcomings the random forest can introduce. They are generally caused by dark regions that highly differ with the typical white color of the blade—a typical example is illustrated in Fig. 4, first row/right side—, where the random forest misplaces the vortex generators as background.

In Fig. 5, we depict the most common cases where our segmentation algorithm could fail. To start with, the BU-Net might occasionally struggle with images of the wind turbine hub, because our dataset has a few instances of the hub. This is the case for the images in the first row. Distinct sources of contamination can also cause the BU-Net to mislead part of the blade with the background. This is the case for the image in the second row and right side. As the blade image borders are not fully captured and these images may contain a broad range of color intensities in the blade region, the hole filling and random forest steps are not sufficient to recover the blade region that has not been identified by the BU-Net.

In addition, an image could include wind turbines at its background or its own tower (see images in the second row).

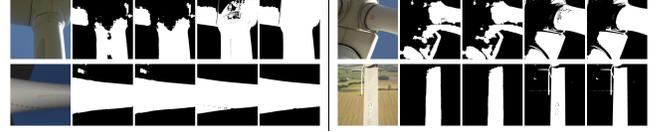


Fig. 5. Failure cases on test images after applying each step of our segmentation algorithm. See caption of Fig. 4.

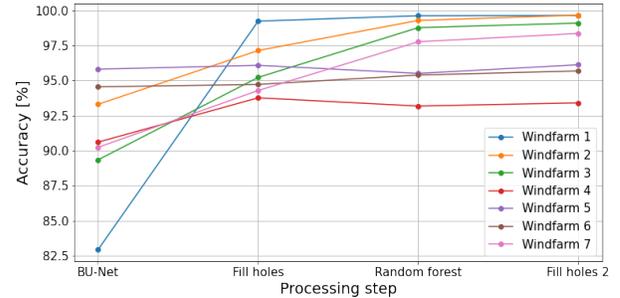


Fig. 6. Accuracy results over the test set for each windfarm. It is included in each step in our method. For readability, the plot does not include every windfarm of the test set.

Ideally, we would like to predict the tower as blade, since it belongs to the turbine itself, but not the turbines in the landscape. This challenging problem is part of our future work.

3.5. Windfarm Dissimilarity

Fig. 6 proves the generability of our approach, demonstrating that images from all the distinct windfarms have a high performance. Notice that this is not the case for the BU-Net without the post-processing steps, as it has really low performance for Windfarm #1. Hence, it is the post-processing steps as a whole which ensures the generability. For instance, hole filling is crucial for Windfarm #1, while random forest is crucial for Windfarm #2 and #3.

4. CONCLUSION

In this paper we have presented a novel segmentation algorithm for wind turbine blade images. To this end, we have proposed a BU-Net model in combination with additional steps based on hole filling and random forest strategies that can solve the problem in an accurate and efficient manner. Our BU-Net exploits a focal loss that is regularized with a contiguity term. The hole filling is simple but effective, and helps to improve the solution after applying every learning model. Finally, our unsupervised random forest is trained on-the-fly capturing the main patterns, like a denoising step. Remarkable quantitative and qualitative results are provided on newly acquired blade images, even when they belong to distinct inspection campaigns and wind farms, validating the capability of our model to generalize properly. Our future work is oriented to extend our model to semantic segmentation where different parts of the wind turbine need to be segmented in the image. Furthermore, we would like to study its impact on facilitating the defect detection task.

5. REFERENCES

- [1] M. Civera and C. Surace, “Non-destructive techniques for the condition and structural health monitoring of wind turbines: A literature review of the last 20 years,” *Sensors*, vol. 22, no. 4, pp. 1627, 2022.
- [2] P. Scarabaggio, S. Grammatico, R. Carli, and M. Dotoli, “Distributed demand side management with stochastic wind power forecasting,” *IEEE Transactions on Control Systems Technology*, vol. 30, no. 1, pp. 97–112, 2022.
- [3] J. Yu, K. Liu, L. Qin, Q. Li, F. Zhao, Q. Wang, H. Liu, B. Li, J. Wang, and K. Li, “DMNet: A new few-shot framework for wind turbine surface defect detection,” *Machines*, vol. 10, no. 6, pp. 487, 2022.
- [4] A.S.M. Shihavuddin, X. Chen, V. Fedorov, A. Nyman Christensen, N. Andre Brogaard Riis, K. Branner, A. Bjorholm Dahl, and R. Reinhold Paulsen, “Wind turbine surface damage detection by deep learning aided drone inspection analysis,” *Energies*, vol. 12, no. 4, pp. 676, 2019.
- [5] L. Zhang, F. Yang, Y. Daniel Zhang, and Y.J. Zhu, “Road crack detection using deep convolutional neural network,” in *International Conference on Image Processing*, 2016, pp. 3708–3712.
- [6] D. Nie, J. Xue, and X. Ren, “Bidirectional pyramid networks for semantic segmentation,” in *Asian Conference on Computer Vision*, 2020.
- [7] B.E. Schäfer, D. Picchi, T. Engelhardt, and D. Abel, “Multicopter unmanned aerial vehicle for automated inspection of wind turbines,” in *2016 24th Mediterranean Conference on Control and Automation (MED)*. IEEE, 2016, pp. 244–249.
- [8] L.C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and 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*, vol. 40, no. 4, pp. 834–848, 2017.
- [9] Xingang Pan, Xiaohang Zhan, Jianping Shi, Xiaoou Tang, and Ping Luo, “Switchable whitening for deep representation learning,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1863–1871.
- [10] J. Wang and T. Lukasiewicz, “Rethinking bayesian deep learning methods for semi-supervised volumetric medical image segmentation,” in *Computer Vision and Pattern Recognition*, 2022, pp. 182–190.
- [11] Y. Meng, H. Zhang, D. Gao, Y. Zhao, X. Yang, X. Qian, X. Huang, and Y. Zheng, “BI-GConv: Boundary-aware input-dependent graph convolution for biomedical image segmentation,” in *British Machine Vision Conference*, 2021.
- [12] M. Yin, Z. Yao, Y. Cao, X. Li, Z. Zhang, S. Lin, and H. Hu, “Disentangled non-local neural networks,” in *European Conference on Computer Vision*. Springer, 2020, pp. 191–207.
- [13] J.Y. Huang and J.J. Ding, “Generic image segmentation in fully convolutional networks by superpixel merging map,” in *Asian Conference on Computer Vision*, 2020.
- [14] L.C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, “Encoder-decoder with atrous separable convolution for semantic image segmentation,” in *European Conference on Computer Vision*, 2018, pp. 801–818.
- [15] Hang Zhang et al., “Resnest: Split-attention networks,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 2736–2746.
- [16] 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*, 2015, pp. 234–241.
- [17] Ali Hatamizadeh, Ziyue Xu, Dong Yang, Wenqi Li, Holger Roth, and Daguang Xu, “Unetformer: A unified vision transformer model and pre-training framework for 3d medical image segmentation,” *arXiv preprint arXiv:2204.00631*, 2022.
- [18] P. Soille, *Morphological image analysis: principles and applications*, Springer Science & Business Media, 1999.
- [19] L. Breiman, “Random forests,” *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [20] T.Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” in *International Conference on Computer Vision*, 2017, pp. 2980–2988.
- [21] P. M. Ferreira, F. Marques, J. S. Cardoso, and A. Rebelo, “Physiological inspired deep neural networks for emotion recognition,” *IEEE Access*, vol. 6, pp. 53930–53943, 2018.
- [22] D.P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.