XMNet: XGBoost with Multitasking Network for Classification and Segmentation of Ultra-Fine-Grained Datasets

Ramy Farag
ViGIR Lab, EECS dept.
University of Missouri-Columbia
Columbia, MO, USA
rmf3mc@missouri.edu

Jacket Demby's
ViGIR Lab, EECS dept.
University of Missouri-Columbia
Columbia, MO, USA
udembys@missouri.edu

Muhammad Arifuzzaman
Division of Plant Science and Technology
University of Missouri-Columbia
Columbia, MO, USA
marifuzzaman.111@gmail.com

Guilherme N. DeSouza
ViGIR Lab, EECS dept.
University of Missouri-Columbia
Columbia, MO, USA
desouzag@missouri.edu

***Abstract*—Classification and segmentation using ultra-fine-grained datasets can be challenging due to the small nuances between adjacent classes. This problem can be exacerbated by the fact that variations within classes can be much larger than other variations between classes. Some approaches have resorted to attention mechanisms that focus on the source or the properties of the features that cause these minor changes in samples between or within classes. In some cases, the attention mechanism can be derived from spatial, temporal, modal, or other types of features in the dataset. Sometimes, attention can be drawn from external sources such as the shape of the object, its skeleton, contour, etc. Finally, some approaches use completely independently extracted information to guide the attention mechanism in a supervised fashion (privileged information, guided-attention, etc). In this paper, we claim that in the context of ultra-fine datasets with a small number of samples, a simple attention mechanism can improve the classification results. Moreover, the same simple attention mechanism can be employed in a backbone topology for the segmentation of the same information that would otherwise be used to guide the attention mechanism in other methods. In other words, unlike the state-of-the-art model for ultra-fine-grained classification of, for example, plant leaves datasets, which uses segmentation masks to guide its attention mechanism, our proposed network can simultaneously provide a classification label and a segmentation mask. The XGBoost algorithm was applied to the attention-modulated feature map for classification, and the Optuna hyperparameter optimization framework was used to tune XGBoost. Three state-of-the-art methods were compared against ours using three benchmark datasets, and our model, XMNet, achieved the best results for the vein segmentation task. For the classification part, our network achieved comparable performance with respect to two state-of-the-art as well as various other more traditional methods.**

***Index Terms*—ultra-fine-grained visual categorization, neural network, semantic segmentation, attention, multitasking, and gradient boosting.**

I. Introduction

Deep learning models, which are deep convolutional neural networks (CNNs) and transformers based models, achieved comparable performance to humans on different large-scale image classification datasets [8], [36]. These CNNs and transformers combine low, mid, and high-level features with classifiers. The variety of feature levels is enhanced by the number of layers, or the depth, of the network [14]. Many studies have shown that the depth of the network is significant for its performance [15], [17], [26], [28]. That allowed CNNs and transformers to be the state-of-the-art models for object recognition [8]. However, these models need large datasets to be able to extract representations that model the differences between classes. Deep learning models are required to produce close representations for instances belonging to the same class, ensuring a high degree of similarity in intra-class representations. Conversely, for instances that are categorized under different labels, these models should generate representations that are distinctly separated. Unfortunately, for small datasets, these models usually memorize the images, leading to an overfitting problem.

Despite their superior performance on different benchmarks, these models do not perform as well on fine-grained and ultra-fine-grained visual classification datasets [21] [2] [22] [38]. Especially for the ultra-fine-grained datasets, that is due to the existence of many classes; each has a small number of instances, and arguably, there are no visual appearance differences between some of the classes, especially in the ultra-fine-grained datasets. Figure 1, contains different classes of the SoyCultivarVein, BtfPIS, and HainanLeaf datasets, respectively. As shown in Figure 1, each dataset has several different cultivars with minimal, if any, visual differences. For example, the first row of Figure 1 has three different categories of soy leaves. The visual distinctions are so subtle that they may even challenge human experts.

Contrary to traditional classification tasks that typically focus on identifying objects from distinct species, such as cats, dogs, aircraft, and cars, the classification methods for fine-grained visual categorization need to be able to find the small differences between the categories in each dataset, such as the BtfPIS dataset. For this type of task, the categories are within the same species. However, they are less challenging to differentiate between than categories in the ultra-fine-grained datasets. For ultra-fine-grained visual categorization, it is a harder task because there are small visual inter-class variances [21], such as the SoyCultivarVein and HainanLeaf datasets.

A number of methods have been developed to address this problem. These techniques involve either adding an additional supervisory signal or an external discriminative input. One technique uses multiple networks to divide the classification task into two parts. The first part is to locate discriminative regions in images. Secondly, it uses these discriminative regions with the input image to classify [18]. Another technique used the segmentation mask, not only as a discriminative region but to guide the network to pay more attention to the regions highlighted by this mask [33].

Leaf vein segmentation in ultra-fine-grained datasets like SoyCultivarVein and HainanLeaf is crucial, as vein characteristics such as vein density significantly influence plant productivity and physiological processes [4], [9]. This is particularly relevant for understanding traits like drought tolerance by finding the correlation between vein characteristics and genetic factors. Genetic factors are hypothesized to control vein characteristic variations within species [10], making it important for the identification of superior genotypes. Effective leaf vein segmentation is essential for quantifying variations within cultivars, aiding in the development of crop varieties with improved yield and stress resilience.

In this paper, a network is developed that does not take either discriminative parts or additional supervisory signals. The classification module of the network is trained solely with the actual classification labels, without the use of segmentation masks to guide its attention mechanism. However, it provides both a classification label and a segmentation mask. It is tested on three benchmarks; one is fine-grained, and the other two are ultra-fine-grained datasets. This network gives better mean intersection over union and classification accuracy than some of the state-of-the-art networks. To the best of our knowledge, this is the first model to provide classification labels and segmentation masks for Ultra-Fine Grained Datasets simultaneously. Our model has three main modules, an encoder-decoder module, an attention module, and a classifier with a hyperparameters tuning algorithm, as shown in Figure 2. In Figure 2, the green circle represents the same above attention mechanism. We have made the code available in the dedicated GitHub repository associated with this paper https://github.com/rmf3mc/xmnet.

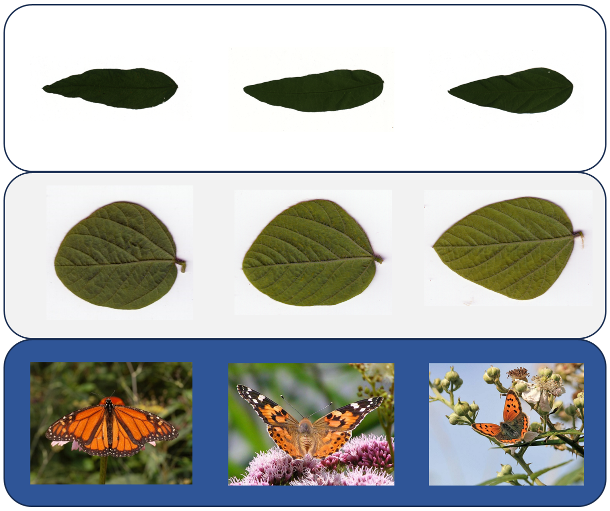

Fig. 1: Samples from SoyCultivarVein, HainanLeaf (HL), and BtfPIS datasets. Each row represents one of these datasets respectively, and each column represents a different class

II. Method

Our proposed method comprises three modules: an encoder and a decoder module for feature extraction and image segmentation, an attention module to enhance relevant features and suppress irrelevant ones, and lastly, the Optuna framework for hyperparameter optimization of XGBoost for classification.

A. Enoder-Decoder

Many studies have demonstrated the effectiveness of using pre-trained models for downstream tasks [5], [19]. In our experiments, we explored a variety of pre-trained models, including DenseNet [17], ResNet [14], and MobileNet [15]. For our final model, we choose to employ a pre-trained DenseNet161 model as a backbone for extracting the final feature map for classification F_4 and, in addition, the multi-scale feature maps from different layers of the backbone F_{0-4} for the decoder's input. Each feature map captures different levels of detail, with deeper layers typically representing more abstract or global features and shallower layers capturing more detailed spatial information. Such that the F_0 feature map has the full scale and F_4 has the one-sixteenth of the full scale.

For the decoder, we employed a simple U-Net decoder [23]. Its main operation includes enlarging the feature map through a twofold upsampling layer, which utilizes bilinear interpolation. This interpolation method calculates a weighted average based on the closest 2×2 neighborhood of known pixel values around the unknown pixel. Finally, the feature map at the top layer of the decoder undergoes convolution with a 1×1 convolutional layer. This step maps each component of the feature vector in the final feature map to a single value to construct the final pixel value in the segmentation mask.

B. Attention

The spatial attention module in XMNet is identical to those used in [33], [21], and [35]. However, in contrast to [33] and

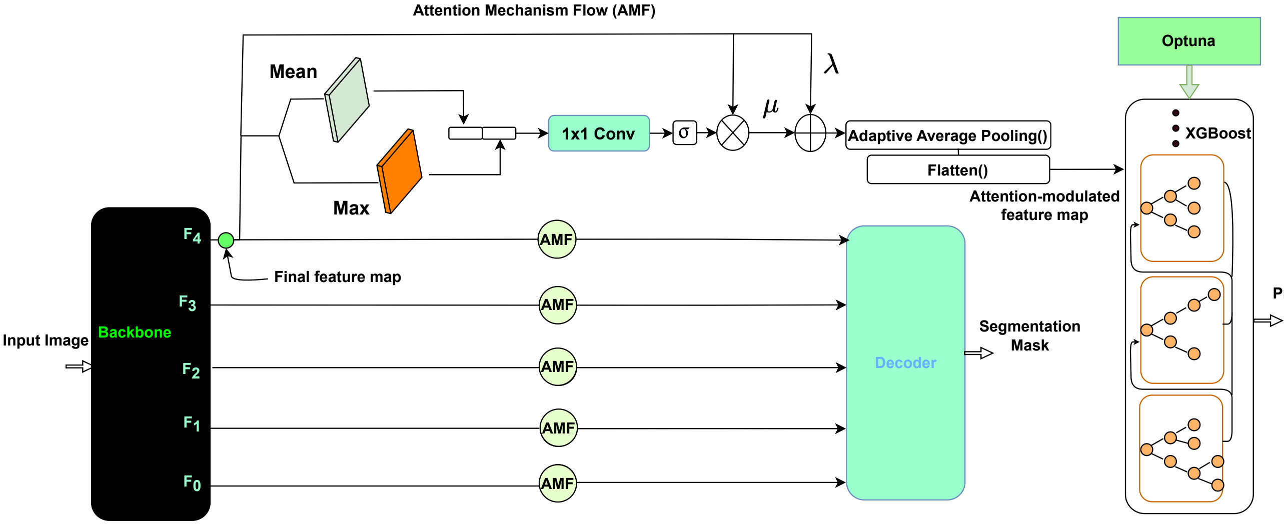

Fig. 2: Schematic diagram of the XMNet modules

[27], our attention module is not guided using a segmentation mask. In other words, the classification part of our model is trained without any additional supervisory signals. It is only trained using the classification labels in an end-to-end manner. The attention module processes the backbone's feature maps to produce a corresponding weight map. This weight map is then applied in a spatially specific manner to the feature maps. As a result, the network can more effectively concentrate on the most pertinent spatial regions within the feature map, enhancing the model's performance [27].

The backbone of our model produces five feature maps, denoted as F_{0-4}. Each of these feature maps is input into the attention module. Let's represent the k-th feature map as $F_k \subseteq \mathbb{R}^{H_k \times W_k \times C_k}$, where H_k, W_k, and C_k correspond to the height, width, and number of channels of F_k, respectively. To process each F_k, we apply mean and maximum operations. These operations compute the mean attention mask $Amean_k$ and the maximum attention mask $Amax_k$ of F_k, they are calculated as the following:

$$Amean_k^{i,j} = \frac{1}{C}\sum_{c=1}^{C} F_{\mathrm{k}}^{i,j}(c), \tag{1}$$

$$Amax_k^{i,j} = F_{\mathrm{k}}^{i,j}(\underset{c}{\mathrm{argmax}}\ F_{\mathrm{k}}^{i,j}(c)), \tag{2}$$

where i, j are the coordinates in the attention map, and c is the c^{th} channel of the image feature maps.

These masks will have spatial regions that are more informative and discriminative for the model to classify the given input. These masks are then used for computation of the final attention mask A_{final_k}, as shown in the following:

$$A_{final_k} = \sigma(f_{1\times1}(Amean_k, Amax_k)) \tag{3}$$

Where $f_{1\times1}$ is a 1x1 convolution, and σ is a sigmoid function applied to the output of the 1x1 convolution. The 1x1 convolution is used to weigh the mean and maximum attention masks according to their importance to the final prediction. Each channel of the attended feature map $F_{\mathrm{att}_k}(c)$, is then calculated by element-wise multiplication of the same channel of the feature map $F_{\mathrm{k}}(c)$ and the final attention mask A_{final_k}.

$$F_{\mathrm{att}_k}(c) = F_{\mathrm{k}}(c) * A_{final_k}, \tag{4}$$

Finally, the final attention-modulated feature map F_{final_k} of the feature map F_{k} is then the product of weighted element-wise addition of the attended feature map F_{att_k} and the feature map F_{k}, as shown in the following.

$$F_{\mathrm{final}_k} = \lambda F_{\mathrm{k}} + \mu F_{\mathrm{att}_k}, \tag{5}$$

The weights λ and μ are hyper-parameters to weigh feature map F_{k} and final attended feature map F_{att_k} according to their contribution [12], in this paper both set to $\frac{1}{2}$ [33].

C. XGBoost with Optuna

Often, enhancements in state-of-the-art image classification benchmarks are achieved not just through new methods, but also through refined configurations of existing models [30]. For example, EfficientNet is developed by finding the balance between network depth, width, and resolution [29]. By optimizing those hyper-parameters, EfficientNet outperformed ResNet, DenseNet and MobileNet [29].

Recent approaches have increasingly integrated various machine learning methods with deep neural networks, capitalizing on the superior ability of these techniques to handle high-dimensional data [11], [24], [31]. One of the most popular machine learning methods is the gradient tree boosting [20]. It has been used in many standard classification benchmarks, and it has become state-of-the-art in many of them. XGBoost is a scalable machine learning system for tree boosting, which is a gradient tree boosting algorithm with several enhancements in terms of computational speed, performance, and usability. Moreover, XGBoost has won challenges by combining it with neural nets [6]. At every iteration, XGBoost gradually improves compared to the previous iteration. Thus, in this paper, we used the XGBoost algorithm atop deep learning models's features.

XGBoost requires careful tuning of several hyperparameters to achieve optimal performance. These parameters include the type of booster, step size shrinkage to prevent overfitting, minimum loss reduction for further partitioning on a leaf node of the tree, and the L2 regularization term on weights. Therefore, hyper-parameter optimization should be considered an essential external loop in the learning process [3].

In our approach, we employed Optuna, a hyperparameter optimization framework [1], utilizing a tree-structured Parzen estimator approach sampler [3] to search for the optimum parameters for XGBoost. During the training process, we continuously input the training loss from each epoch of XGBoost's objective funtion into Optuna. In the context of the objective function for XGBoost, we define several terms. The simplified objective function at step t, denoted as $\mathcal{L}^{(t)}$, is given by

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} [g_i f_t(\mathbf{x}_i) + \frac{1}{2} h_i f_t^2(\mathbf{x}_i)] + \Omega(f_t), \tag{6}$$

where g_i and h_i are the first and second order gradient statistics on the loss function for the i^{th} instance. These gradients are calculated as follows:

$$g_i = \frac{\partial}{\partial \hat{y}^{(t-1)}} l(y_i, \hat{y}^{(t-1)}), \tag{7}$$

$$h_i = \frac{\partial^2}{\partial (\hat{y}^{(t-1)})^2} l(y_i, \hat{y}^{(t-1)}). \tag{8}$$

Here, $f_t(\mathbf{x}_i)$ represents the prediction of the t^{th} tree for the i^{th} sample. The regularization term $\Omega(f_t)$, which is added to prevent overfitting by penalizing the complexity of the model, is expanded as

$$\Omega(f_t) = \gamma T + \frac{1}{2} \lambda \sum_{j=1}^{T} w_j^2, \tag{9}$$

where γ is a parameter that controls the complexity of the model with respect to the number of leaves T, λ is the L2 regularization term, and w_j are the leaf weights of the t^{th} tree. This method facilitates a more efficient and broadly applicable tuning of the model's parameters, enhancing both performance and generalizability.

III. Results

A. Datasets

All three datasets have both image- and pixel-level labels. **SoyCultivarVein Dataset**. The SoyCultivarVein dataset [37] is a publicly available collection of leaf images encompassing two hundred different cultivars, each with six individual samples. This amounts to a total of 1200 images in the dataset. All these categories are derived from the same species, which makes the dataset particularly challenging due to the considerable resemblance among the categories. For the purpose of classification tasks, the dataset is divided into a training set and a test set, following a 2:1 distribution ratio.

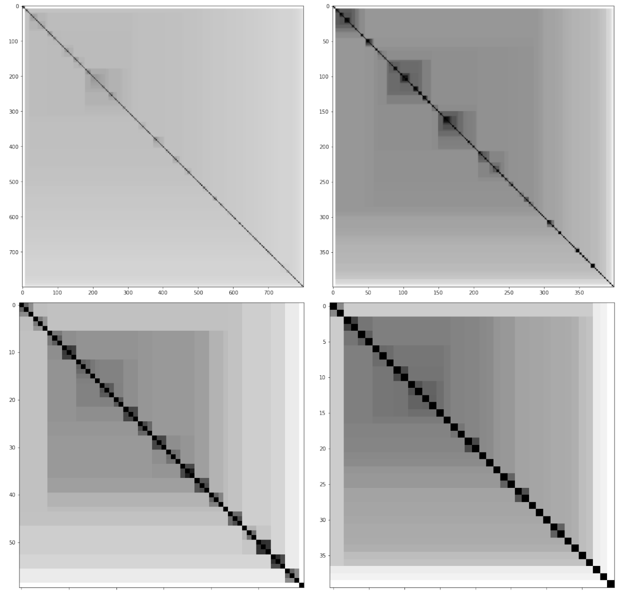

Fig. 3: iVAT algorithm on the Attended features of DenseNet161 model. The top row is the output of iVAT given the training and testing attended features of SoyCultivarVein, and the second row of HainanLeaf

HainanLeaf Dataset. The HainanLeaf [39], comprises one hundred images distributed over twenty distinct categories. Each category is represented by five individual samples. Similar to the SoyCultivarVein dataset, this dataset is also divided into a training set and a test set with a ratio of 3:2, respectively.

BtfPIS dataset. The Butterfly Patchy Image Structure Dataset [32], commonly known as the BtfPIS dataset, encompasses a collection of five hundred images, evenly distributed across ten categories, with each category containing fifty images. One-fifth of the total images are designated for training the model, while the remaining four-fifths of the images are set aside for testing purposes.

B. Classification Results

In Figure 3, we used the improved Visual Assessment of Cluster Tendency (iVAT) Algorithm to test the clustering tendency of the training and testing sets of SCV and HL [13]. As shown in Figure3, the model classifies the SCV dataset by identifying clusters with similar features. This suggests a potential lack of visual distinction between classes within these clusters. Conversely, with the HL dataset, the model demonstrates better generalization, successfully distinguishing the features representative of each class.

As mentioned earlier, our model incorporated the XGBoost algorithm with a deep-learning model and Optuna. In this paper, we experimented with different pre-trained models used as a backbone before choosing the best-suited one for the ultra-fine-grained datasets. These models are DenseNet161,

Method	Technique		
Hand-crafted Features	MDM [16]	13.4	
	DBCSR [7]	14.1	
	MORT [39]	28.5	
CNN	Backbone	Architecture	
		MGA	Orig.
	DenseNet161	35.5	33.0
	DenseNet121	33.5	30.0
	Resnet50	29.25	31.25
	Resnet34	26.5	25.25
	Resnet18	27.0	25.75
	MobilenetV2	28.25	29.25
	MobilenetV3	31.0	28.25
GB	DenseNet161	6.25	
AdaBoost	DenseNet161	1.50	
RF	DenseNet161	27.25	
XGBoost	DenseNet161	26.0	
PAC	DenseNet161	33.75	
SVM	DenseNet161	24.55	
KNN	DenseNet161	26.33	
Ours	DenseNet161	34.75	

TABLE I: The accuracy in % of the experimented methods on the test sets of SoyCultivarVein (SCV) dataset

Dataset	Backbone	Architecture		
		XMNet	MGA	Orig.
SCV	DenseNet161	34.75	35.5	33.0
HL	DenseNet161	75.0	77.5	77.5
BtfPIS	DenseNet161	89.25	87.0	86.75

TABLE II: The accuracy in % of the experimented CNN based models on the test sets of SoyCultivarVein (SCV), HainanLeaf (HL), and BtfPIS datasets

DenseNet121, Resnet50, Resnet34, Resnet18, MobileNetV2, and MobileNetV3.

We compared our model with a number of other methods, including convolution neural network based models, hand crafted features, and other machine learning methods atop Convolution Neural Network models. The hand crafted based methods are reported from [33]. The convolution neural network based models are Mask Guided Attention (MGA) and its corresponding original models [33]. And the hand crafted features are deformation based curved shape representation (DBCSR) [7], multiscale distance matrix for fast plant leaf recognition (MDM) [16] and patchy image structure classification using multi-orientation region transform (MORT) [39]. The other machine learning methods are gradient boosted decision trees (GBDT) [20], Adaboost [25], random forest (RF), XGBoost [6], passive aggressive classifier (PAC), support vector machine (SVM) and k-nearest neighbor (KNN).

In TABLE I, we reported the accuracy (%) of all the methods we tested on the SoyCultivarVein (SCV) test dataset. While, In TABLE II, we reported the accuracy (%) of all the Convolution Neural Network based models we tested on the three datasets. As shown in both tables, our approach outperforms both machine learning, hand crafted features based methods and gives similar results to the Convolution Neural Network state-of-the-art based models.

	SCV	HainanLeaf
HMSANet	52.53	41.31
Unet3+	50.56	4.13
InternImage	62.86	52.65
Ours	**65.96**	**59.94**

TABLE III: The semantic segmentation results (mIoU) on the test sets of SoyCultivarVein (SCV), HainanLeaf (HL) and BtfPIS datasets.

C. Segmentation Results

After training the model on classification, we halted updating the model backbone's parameters and attached and trained a decoder to segment the given input images. We tested a Unet3+ decoder with the bare features from the backbone and another Unet decoder with the attended features, as shown in Figure 2. The mean intersection over Union (mIoU) results are reported in Table III. We reported the HMSANet results from [33] for additional comparison. Moreover, we trained and evaluated the IntermImage model on the three datasets [34]. As shown in Table III, our model outperforms the other models on the ultra-fine-grained datasets.

IV. Conclusions

In conclusion, in this paper, we developed a new method to, namely XMNet, that provide a dual output of classification labels and segmentation masks without relying on auxiliary supervision signals. XMNet simplifies the computation process. As samples for ultra-fine-grained are expensive to acquire, thus our design focused on using simple operations to prevent overfitting as much as possible. The model shows its ability to discern minute appearance variances among different categories that are extremely hard to be noticed by experts if these variabilities exist and handle the intrinsic variability within the same category—tasks. We utilized the XGBoost algorithm atop the final feature that was refined using an attention heatmap. The incorporation of Optuna for hyperparameter optimization of XGBoost further refined the learning process, ensuring that the model's parameters are finely tuned to the dataset at hand. The empirical evaluation of XMNet across benchmark datasets—the ultra-fine-grained SoyCultivarVein and HainanLeaf demonstrates its robustness in vein segmentation, which is a critical factor in leaf morphology and genetic studies. Even with a frozen encoder post the classification training, XMNet outperforms contemporary deep learning models in ultra-fine-grained datasets. For classification, XMNet's performance is on par with that of state-of-the-art convolution-based neural networks. Notably, it surpasses methods that rely on hand-crafted features, highlighting the effectiveness of deep learning approaches.

Acknowledgements

Computational resources for this research have been supported by the NSF National Research Platform, as part of GP-ENGINE (award OAC #2322218).

References

[1] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2623–2631, 2019.

[2] Ardhendu Behera, Zachary Wharton, Pradeep RPG Hewage, and Asish Bera. Context-aware attentional pooling (cap) for fine-grained visual classification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 929–937, 2021.

[3] James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. *Advances in neural information processing systems*, 24, 2011.

[4] C Kevin Boyce, Tim J Brodribb, Taylor S Feild, and Maciej A Zwieniecki. Angiosperm leaf vein evolution was physiologically and environmentally transformative. *Proceedings of the Royal Society B: Biological Sciences*, 276(1663):1771–1776, 2009.

[5] Shoufa Chen, Chongjian Ge, Zhan Tong, Jiangliu Wang, Yibing Song, Jue Wang, and Ping Luo. Adaptformer: Adapting vision transformers for scalable visual recognition. *Advances in Neural Information Processing Systems*, 35:16664–16678, 2022.

[6] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[7] Girum Getachew Demisse, Djamila Aouada, and Björn Ottersten. Deformation based curved shape representation. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1338–1351, 2017.

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[9] Linda L Flora and Monica A Madore. Significance of minor-vein anatomy to carbohydrate transport. *Planta*, 198(2):171–178, 1996.

[10] Shangjing Guo, Mingyi Zhu, Jianjun Du, Jinglu Wang, Xianju Lu, Yu Jin, Minggang Zhang, Xinyu Guo, and Ying Zhang. Accurate phenotypic identification and genetic analysis of the ear leaf veins in maize (zea mays l.). *Agronomy*, 13(3):753, 2023.

[11] Taher Hajilounezhad, Rina Bao, Kannappan Palaniappan, Filiz Bunyak, Prasad Calyam, and Matthew R Maschmann. Predicting carbon nanotube forest attributes and mechanical properties using simulated images and deep learning. *npj Computational Materials*, 7(1):134, 2021.

[12] Mohammed Hassanin, Saeed Anwar, Ibrahim Radwan, Fahad S Khan, and Ajmal Mian. Visual attention methods in deep learning: An in-depth survey. *arXiv preprint arXiv:2204.07756*, 2022.

[13] Timothy C Havens and James C Bezdek. An efficient formulation of the improved visual assessment of cluster tendency (ivat) algorithm. *IEEE Transactions on Knowledge and Data Engineering*, 24(5):813–822, 2011.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. corr abs/1512.03385 (2015), 2015.

[15] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[16] Rongxiang Hu, Wei Jia, Haibin Ling, and Deshuang Huang. Multiscale distance matrix for fast plant leaf recognition. *IEEE transactions on image processing*, 21(11):4667–4672, 2012.

[17] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[18] Shaoli Huang, Zhe Xu, Dacheng Tao, and Ya Zhang. Part-stacked cnn for fine-grained visual categorization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1173–1182, 2016.

[19] Lingbo Liu, Jianlong Chang, Bruce XB Yu, Liang Lin, Qi Tian, and Chang-Wen Chen. Prompt-matched semantic segmentation. *arXiv preprint arXiv:2208.10159*, 2022.

[20] Alexey Natekin and Alois Knoll. Gradient boosting machines, a tutorial. *Frontiers in neurorobotics*, 7:21, 2013.

[21] Zicheng Pan, Xiaohan Yu, Miaohua Zhang, and Yongsheng Gao. Mask-guided feature extraction and augmentation for ultra-fine-grained visual categorization. In *2021 Digital Image Computing: Techniques and Applications (DICTA)*, pages 1–8. IEEE, 2021.

[22] Jo Plested, Xuyang Shen, and Tom Gedeon. Non-binary deep transfer learning for image classification. *arXiv preprint arXiv:2107.08585*, 2021.

[23] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015.

[24] Kaveh Safavigerdini, Koundinya Nouduri, Ramakrishna Surya, Andrew Reinhard, Zach Quinlan, Filiz Bunyak, Matthew R Maschmann, and Kannappan Palaniappan. Predicting mechanical properties of carbon nanotube (cnt) images using multi-layer synthetic finite element model simulations. In *2023 IEEE International Conference on Image Processing (ICIP)*, pages 3264–3268. IEEE, 2023.

[25] Robert E Schapire. Explaining adaboost. In *Empirical Inference: Festschrift in Honor of Vladimir N. Vapnik*, pages 37–52. Springer, 2013.

[26] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[27] Chunfeng Song, Yan Huang, Wanli Ouyang, and Liang Wang. Mask-guided contrastive attention model for person re-identification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1179–1188, 2018.

[28] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[29] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.

[30] Mingxing Tan, Ruoming Pang, and Quoc V. Le. Efficientdet: Scalable and efficient object detection. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10778–10787, 2019.

[31] Imad Eddine Toubal, Linquan Lyu, Dan Lin, and Kannappan Palaniappan. Single view facial age estimation using deep learning with cascaded random forests. In *Computer Analysis of Images and Patterns: 19th International Conference, CAIP 2021, Virtual Event, September 28–30, 2021, Proceedings, Part II 19*, pages 285–296. Springer, 2021.

[32] Josiah Wang, Katja Markert, Mark Everingham, et al. Learning models for object recognition from natural language descriptions. In *BMVC*, volume 1, page 2. Citeseer, 2009.

[33] Jun Wang, Xiaohan Yu, and Yongsheng Gao. Mask guided attention for fine-grained patchy image classification. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 1044–1048. IEEE, 2021.

[34] W Wang, J Dai, Z Chen, Z Huang, Z Li, X Zhu, X Hu, T Lu, L Lu, H Li, et al. Internimage: Exploring large-scale vision foundation models with deformable convolutions. arxiv. *arXiv preprint arXiv:2211.05778*, 2022.

[35] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018.

[36] Mingyuan Xin and Yong Wang. Research on image classification model based on deep convolution neural network. *EURASIP Journal on Image and Video Processing*, 2019:1–11, 2019.

[37] Xiaohan Yu, Yongsheng Gao, Shengwu Xiong, and Xiaohui Yuan. Multiscale contour steered region integral and its application for cultivar classification. *IEEE Access*, 7:69087–69100, 2019.

[38] Xiaohan Yu, Jun Wang, Yang Zhao, and Yongsheng Gao. Mix-vit: Mixing attentive vision transformer for ultra-fine-grained visual categorization. *Pattern Recognition*, 135:109131, 2023.

[39] Xiaohan Yu, Yang Zhao, Yongsheng Gao, Shengwu Xiong, and Xiaohui Yuan. Patchy image structure classification using multi-orientation region transform. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 12741–12748, 2020.