A 1d convolutional network for leaf and time series classification

Dongyang Kuang

Department of Mathematics and Statistics, University of Ottawa, CA

Abstract

In this paper, a 1d convolutional neural network is designed for classification tasks of leaves with centroid contour distance curve (CCDC) as the single feature. With this classifier, simple feature as CCDC shows more discriminating power than people thought previously. The same architecture can also be applied for classifying 1 dimensional time series with little changes. Experiments on some benchmark datasets shows this architecture can provide classification accuracies that are higher than some existing methods.

1. Introduction

Vast amount of plant species exists on earth, according to [1, 2], there are about 220,000 to 420,000 different species just for flowering plants alone. The large number of plant species, together with the fact that large in-species variations and small cross-species variations make it a difficult and tedious work for identifying them by human, particularly for non-experts. As with the fast development in techniques of machine learning and deep learning methodologies as well as the growing power of computation, automatic recognition with these species become a more and more natural solution.

From a descriptive point of view, plant identification are traditionally based on observations of its organs, such as flowers, leaves, seeds, etc. A large portion of species information is contained in leaves. It also appears for a considerable amount of time during plants' life cycle. This brings benefits for database construction. Traditionally, features from leaves can be roughly divided into three categories: shape, color and texture. Shape descriptors (especially the the contour) usually are more robust compared to the other two. For a single leaf, color descriptors may vary depending on lighting conditions, image format, etc. Texture descriptors can vary if there are worm holes on the leaf... Another advantage of a shape descriptor is that features like centroid center contour curve (CCDC) can be converted to time

Preprint submitted to *December 11, 2018*

series [3], hence techniques in time series classification such as dynamic time warping (DTW) [4] can be applied. On the other hand, techniques that are suitable for leaf classification with this kind of shape descriptor can be easily modified to general time series classification tasks, which will result in a broader field of applications.

Despite the differences of features, traditional classifiers in applications usually includes: support vector machines (SVM), k nearest neighbors (kNN), random forest ... Artificial neural networks, especially convolutional neural networks (CNN) [5] are not commonly seen in the field, though they have proven to be very effective tools in the field of computer vision and pattern recognition. In this paper, discussions are focused on features that are based on leaf shapes and argues that simple shape feature actually contains more discriminating power than people usually think, if an effective classifier such convolutional neural networks are used. The rest of the paper is organized as below: Section 2 gives some related work using shape features for classification. Section 3 presents the design of a 1d convolutional network as a classifier that can also be directly applied to tasks of classifying 1 dimensional time series. Section 4 tests the performance of this classifier on some benchmark data sets.

2. Related Work

Effort for developing classification tools can generally be divided into two parts: extracting features that are more discriminative and designing more effective classifiers.

On the side of shape features, they can be extracted based on botanical characteristics [6, 7]. These features may include: Aspect Ratio, Rectangularity, Convex Area, Ratio, Convex Perimeter Ratio, Sphericity, Circularity, Eccentricity, Form Factor, etc. [8] discussed some other features applied on leave shapes and introduced two new multiscale triangle representations. There are also a lot of other work done with more in-depth design aiming for general shapes than just leaves. [9] defines inner distance of shape contours to build shape descriptors. [10] develops the visual descriptor called CENTRIST (CENsus TRansform hISTogram) for scene recognitions, it get good performance when applied to leave images. Authors of [3] uses the transformation form shape contours to 1 dimensional time series and present the method of shapelet for shape recognition. [11] describes a hierarchical representation for two dimensional objects that captures shape information at multiple levels of resolution for matching deformable shapes. Features coming from different method can be stacked together, these bagged features can usually help provide better performance as discussed in [12].

Among these features used, centroid center contour curve (CCDC) is a feature that is from a relatively easy concept and can be efficiently/conveniently extracted from leaf images. Some early work [13, 14] used it as the single feature or in addition to other features. It was not used (at least not as a single feature) in recent years because people doubt that it may not have enough discriminative power. This paper argues that if a classifier is designed properly, it can reveal more hidden information out of CCDC and provide comparable or better performance when compared to some state-of-art methods mentioned above.

To obtain CCDC representation, one first apply a filter such as a canny filter [15] on the image to obtain the leave contour. For point (x, y) on this contour, its polar coordinates (ρ, θ) is then computed:

$$\rho = \sqrt{(x - x_0)^2 + (y - y_0)^2} \tag{1}$$

$$\theta = \arctan \frac{y - y_0}{x - x_0}. \tag{2}$$

(x_0, y_0) is the image center and can be computed from image moments [16]. Values of ρ then can be sampled on a uniform grid of θ by interpolation. CCDC is obviously translation invariant. It can also be rotation and scale invariant after proper normalization.

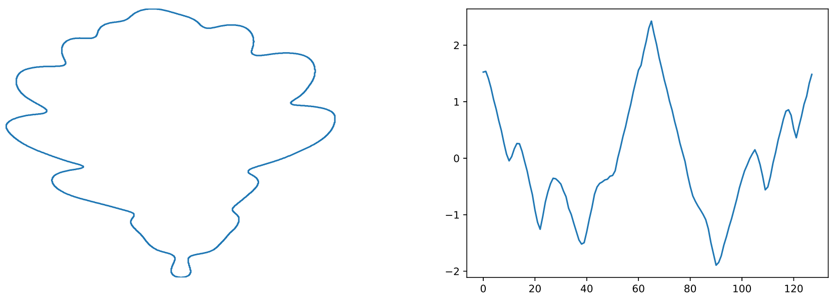

Figure 1: An example of CCDC. Left: Outline of one Quercus leaf; Right: the converted CCDC.

Compared with methods mentioned above which tackles the difficulty in classification by designing complicated hand crafted deep features, convolutional neural networks (CNN) [5] can take simple features as input and automatically abstracts useful features through its early convolutional blocks for later classification tasks [17]. In this way, the difficulty is transferred into heavy computation where modern hardware now can provide sufficient support. It is more straightforward if we apply a CNN directly on leave images combining feature extraction task and classification task together, but this will make a model of unnecessary large size with a lot of parameters and they usually require a lot of data and time to be trained well with more risk of overfitting the data at hand. The key idea of this paper is to take the advantage of convolutional architecture, but apply it on the extracted single 1d CCDC feature to reduce the computational cost.

3. Classifier Design

In order to make proper classification, it is important that the classifier can learn features at different scales together and combine them into classification. Though this can be done by designing complicated hand-crafted features, applying convolutional kernels with different sizes and strides serves as one good option for this purpose. For a typical 1d convolutional mechanism, information flows to the next layer first by a convolutional operation and then processed by an activation function: $Y = f(W * X + B)$, where $*$ denotes the discrete convolution operation between the incoming signal X and a kernel W. A convolutional layer contains several different kernels, computes the convolution between the input and each kernel and then stack their result as its output. Figure 2 gives an illustration of this, the convolutional layer contains several kernels of length 3. During convolution, a sliding window of the same size will slide through the input with certain stride. During each stay of the window, it computes the inner product between the examined portion of input and the kernel itself. For example, when using kernel (3,-1,0) with stride 2 and no bias, the first output is $3\times3+2\times(-1)+4\times0 = 7$ and the second output is $4 \times 3 + 1 \times (-1) + 0 \times 0 = 11$.

Based on this thought, a basic architecture used for classification is designed as in Figure 3. It looks like a naive module from Google's inception network [18] but is built for 1 dimensional input. The input is first processed by convolutional blocks of different configurations which responses to features of different scales. Their outputs are then concatenated together with original input before being fed into latter layers for classification.

In the following experiment section, this network is used in two ways. The first approach is to use it as a classifier allowing informations flow from CCDC feature to species label directly. The other way is to use it as an automatic feature extractor in a "pretrain-retrain" style. During the training phase, the network is first pre-trained to certain extent with earlystopping or a checkpoint at best validating performance. In the testing phase, the model weights are frozen, the top layer is then taken off and its input as pretrained features are fed to a nonlinear classifier such as a SVM or a kNN classifier for final classification. It is like a transfer learning design, but the difference is

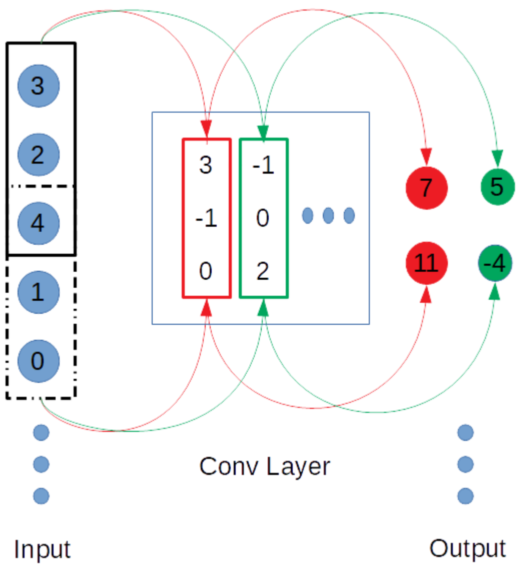

Figure 2: Mechanism of a 1d convolutional layer.

in transfer learning, the model is not trained on the same dataset. The idea is from heuristic that a nonlinear classifier may performance better than the original linear classification performed by the top layer. Experiments done in the next sections shows this (referred as 1dConvNet+SVM) usually will help contribute a little more accuracy to the classification.

4. Experiment Results

4.1. Swedish Leaf

Swedish leaf data set [20] contains leaves that are from 15 species. Within each species, 75 samples are provided. It is an challenging classification task due to its high inter-species similarity [8].

Table 1 lists some existing methods that uses leaf contours for classification. All listed methods in the table use leaf contours in a non-trivial way that involves more in-depth feature extraction than CCDC.

Method	Accuracy	Method	Accuracy
Söderkvist [21]	82.40%	Spatial PACT [10]	90.61%
SC + DP [9]	88.12%	Shape-Tree [11]	96.28%
IDSC + DP [9]	94.13%	TSLA [8]	96.53%

Table 1: Performance of different existing methods on leaf contours.

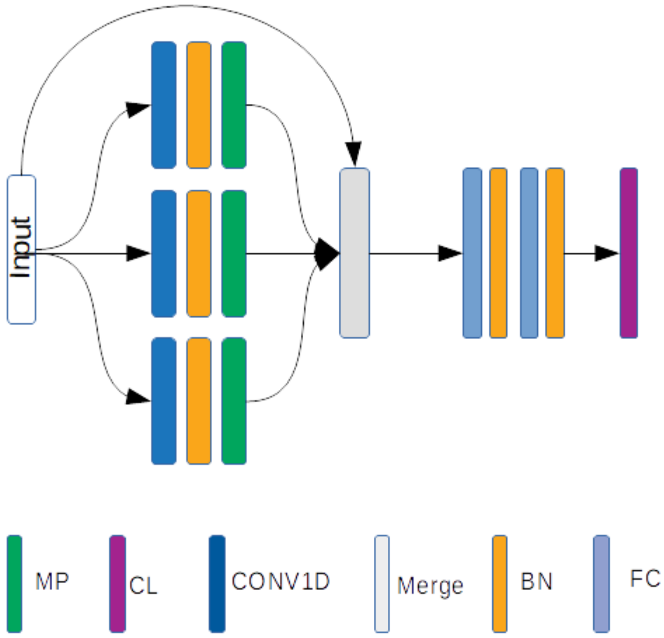

Figure 3: The architecture of the neural network classifier. The right most layer is a classifier layer (CL). It can be a linear classifier, a (kernel) SVM classifier, a knn classifier or other classifiers. The merge layer is simply concatenation of features. Batch normalization (BN) [19] layer can be asserted after the output of convolutional or full connected layer (FC) to help better training. The three convolutional layers are with *different* sizes and strides.

While [8, 9, 10, 11, 21] uses 25 samples randomly selected from each species as the training set and the rest as test. The author decided to use a 10-fold cross validation to evaluate the proposed model in a more robust way. The other reason for this is the convoluational architecture may not be trained sufficiently with 25 samples per species as the training set. The mean performance and the corresponding standard deviation is summarized in Table 2. The actual parameters used are: Convolutional layers {conv1d(16, 8, 4)[1], conv1d(24, 12, 6), conv1d(32, 16, 8)}, Maxpooling layers (MP) are with window size 2 and stride 2, two fully connected layers are of unit 512 and 128, respectively. Relu activations [22] are used in convolutional layers and PRelu [23] activations are used for fully connected layers. To prevent overfitting, Gaussian noise (mean: 0, std: 0.01) layers are placed before each convolutional layer and a dropout layer [24] of intensity 0.5 is inserted before

[1]16 kernels with window size 8 and stride 4.

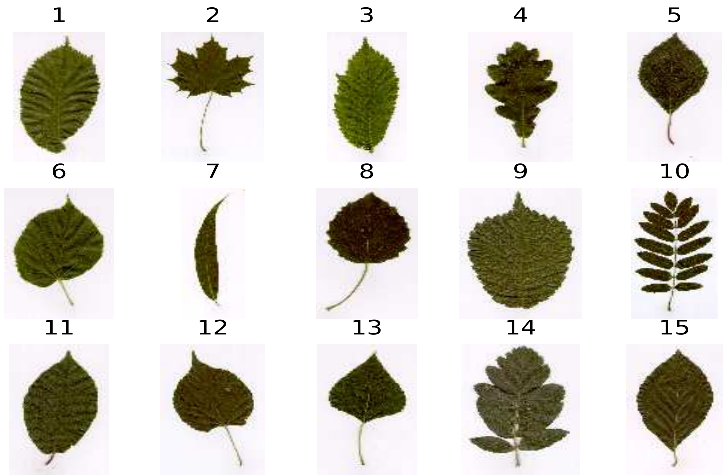

Figure 4: The first sample of each species in the Swedish leaf dataset. 1. Ulmus capinifolia, 2. Acer, 3. Salix aurita, 4. Quercus, 5. Alnus incana, 6. Betula pubescens, 7. Salix alba 'Sericea', 8. Populus tremula, 9. Ulmus glabra, 10. Sorbus aucuparia, 11. Salix sinerea, 12, Populus, 13. Tilia, 14, Sorbus intermedia, 15. Fagus silvatica

the classification layer. The whole model is trained using stochastic gradient descent algorithm with batch size 32, learning rate 0.005 and 10^{-6} as the decay rate. 25 principal components from pretrained features are used if the top classification layer is a SVM. For other details, please check the actual code at [25].

Method	Mean Accuracy	STD	Best	Worst
1d ConvNet	**96.11%**	1.54%	**98.23%**	92.92%
1d ConvNet + 3NN	**94.69%**	1.58%	**96.46%**	91.15%
1d ConvNet + SVM	**97.08%**	1.48%	**99.12%**	94.69%

Table 2: Performance of the 10-fold cross validation using the 1d ConvNet.

The proposed network provides comparable accuracy with top methods listed in Table 1. With a SVM on pretrained features from the network, it is able to provide a better accuracy. A 3NN classifier on the same pretrained features does not give better performance in this experiment.

The UEA & UCR Time Series Classification Repository [26] provides an explicit split of training/test set of this dataset and a list of performances from different time series classification methods, which allows a more direct comparison with the proposed 1d convolutional network. Table 3 lists the best performance reported on the website and results obtained by the proposed 1d ConvNet. The result is obtained by averaging the test accuracy among 5 independent runs with different random states. 20% of the training samples are used as validation for stopping the training process[2]. As seen

Method	Accuracy
COTE[27]	96.67%
1dConvNet	**96.10%**
1dConvNet+3NN	**96.16%**
1dConvNet+SVM	**97.47%**

Table 3: Performance comparison on the explicit training/test split from the UEA & UCR Time Series Classification Repository.

in both comparisons, with top layers replaced by a SVM, the accuracy can be further improved. The reason may be the fact that if the network is already trained properly, information that flows into the top layer is almost linearly separable, hence a nonlinear classifier built on top will help increase the accuracy by correcting some mistakes made by a linear classifier. Figure 5 shows the TSNE embedding [28] with the outputs of the network before the last classification layer from the whole dataset. As one can see in this 2 dimensional feature projection, the 15 classes are almost separable.

4.2. UCI's 100 leaf

UCI's 100 leaf dataset [29] was first used in [12] in support of authors' probabilistic integration of shape, texture and margin features. It has 100 different species with 16 samples per species[3]. As for the feature vector, a 64 element vector is given per sample of leaf. These vectors are taken as a contigous descriptors (for shape) or histograms (for texture and margin). An mean accuracy of 62.13% (with PROP) and 61.88% (with WPROP) was reported by only using the shape feature(CCDC) from a 16-fold validation (10% of training data are hold as validation). The mean accuracy raised up to 96.81% and 96.69% if both three types of features are combined. Following the evaluation of 16-fold validation, the performance of using the 1d ConvNet is summarized in Table 4. For results by combing the 3 features, the author simply concatenates them together to form a 192 dimensional feature vector per sample.

[2]Unless specied otherwise, accuracies recorded in the rest experiments of this paper is obtained with the same way.

[3]One sample's texture feature from the first species is missing, so actually data from the other 99 species is used in this experiment.

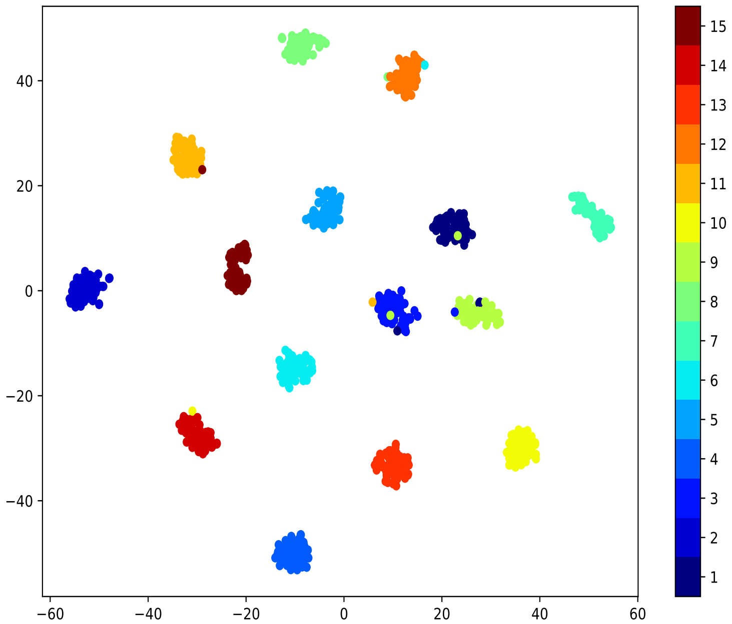

Figure 5: TSNE embedding of the whole dataset using the inputs from the classification layer. The 15 classes are almost linear separable.

Again, the proposed network works better on both kinds of features. The 3-NN with pretrained features from the network did not perform better than the original network. Part of the reason may be because kNN classifier is more sensitive to changes in data and 3 may not be a good choice for k in this dataset which has 99 different classes.

4.3. On some time series Classification

The classifier does not only achieve good performance in classifying different leaves on single CCDC feature, it can also be directly used for classifying 1 dimensional time series data from end to end. In order to demonstrate this, the author selects four different data sets from UEA & UCR Time Series Classification Repository [26]: ChlorineConcentration, InsectWingbeatSound, DistalPhalanXTW and ElectricDevices[4] for test. These data sets comes from different backgrounds with different data sizes and length of feature vectors. A good classification strategy usually requires some prior

[4]Details of these data can be found at the website [26].

Method	CCDC	All 3 features
PROP	62.13%	96.81%
WPROP	61.88%	96.69%
1dConvNet	**73.99% ± 3.72%**	**99.05% ± 0.67%**
1dConvNet+3NN	**73.86% ± 3.66%**	**98.73% ± 1.41%**
1dConvNet+SVM	**77.34% ± 3.55%**	**99.43% ± 0.62%**

Table 4: Comparison of performance on UCI's 100 leaf dataset.

knowledge. With the help of convolutional architecture, the proposed network is able to help reduce such prior knowledge from human. This kind of prior knowledge is "learned" by the network during training. The current best performance reported on the website and performance achieved by this 1d convolutional net are compared in Tabel 5. For all the four datasets, the network's architecture and hyperparameters are the same as previous experiments with no extra hyperparameter tuning[5]. As summarized in Table 5, the proposed network outperforms the reported best methods in terms of mean accuracy.

Dataset	Classes	Best Method Reported		1dConvNet+SVM
ChlorineConcentration	3	90.41%	*SVM(quadratic)*	**99.77%**
InsectWingbeatSound	11	64.27%	*Random Forrest*	**76.61%**
ElectricDevices	7	89.54%	*Shapelet Transform*[30]	**94.34%**
DistalPhalanXTW	6	69.32%	*Random Forrest*	**71.22%**

Table 5: Performance achieved by the proposed 1d convolutional netwrok compared to reported best performance on [26].

5. Conclusion

This paper presents a simple 1 dimensional convolutional network architecture that allows classification tasks of plant leaves on single CCDC feature instead of further extracting more complicated features. The same architecture is directly applicable to classify 1 dimensional time series allowing an end-to-end training without complicated preprocessing of input data. Experiments of this classifier on some benchmark datasets show comparable or better performance than other existing methods.

[5]For the DistalPhalanXTW dataset, the author took 10% of them as validation.

Ackowledgement

The author thanks Prof. Tanya Schmah and Dr. Alessandro Selvitella for their kind help in providing many useful suggestions.

References

[1] R.W.Scotland and A.H.Wortley., "How many species of seed plants are there?," *Taxon*, vol. 52, pp. 101–104, 2003.

[2] R.Govaerts., "How many species of seed plants are there?," *Taxon*, vol. 50, pp. 1085–1090, 2001.

[3] L. Ye and E. Keogh, "Time series shapelets: A new primitive for data mining," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, (New York, NY, USA), pp. 947–956, ACM, 2009.

[4] D. J. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series," in *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining*, AAAIWS'94, pp. 359–370, AAAI Press, 1994.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[6] C. Caballero and M. C. Aranda., "Plant species identification using leaf image retrieval.," *In ACM International Conference on Image and Video Retrieval (CIVR)*, pp. 327–334, 2010.

[7] J. Du, X. Wang, and G. Zhang., "Leaf shape based plant species recognition.," *Applied Mathematics and Computation*, vol. 185, pp. 883–893, 2007.

[8] S. Mouine, I. Yahiaoui, and A. Verroust-Blondet, "A shape-based approach for leaf classification using multiscaletriangular representation," in *Proceedings of the 3rd ACM Conference on International Conference on Multimedia Retrieval*, ICMR '13, (New York, NY, USA), pp. 127–134, ACM, 2013.

[9] H. Ling and D. W. Jacobs, "Shape classification using the inner-distance," *IEEE transactions on Pattern Analysis and Machine Intelligence*, vol. 29, pp. 286–299, 2007.

[10] J. Wu and J. M. Rehg, "Centrist: A visual descriptor for scene categorization," *IEEE transactions on Pattern Analysis and Machine Intelligence*, vol. 33, pp. 1489–1501, 2011.

[11] P. Felzenszwalb and J. Schwartz, "Hierarchical matching of deformable shapes," *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[12] C. Mallah, J. Cope, and J. Orwell, "Plant leaf classification using probabilistic integration of shape, texture and margin features.," *Signal Processing, Pattern Recognition and Applications*, vol. 8, pp. 679–714, 2013.

[13] Z. Wang, Z. Chi, D. Feng, and Q. Wang., "Leaf image retrieval with shape features. advances in visual information systems.," *Signal Processing, Pattern Recognition and Applications*, pp. 41–52, 2000.

[14] Y. Shen, C. Zhou, and K. Lin., "Leaf image retrieval using a shape based method.," *Artificial Intelligence Applications And Innovations*, pp. 711–719, 2005.

[15] J. Canny, "A computational approach to edge detection," *IEEE transactions on pattern analysis and machine intelligence*, vol. 8, pp. 679–714, 1986.

[16] T. H. Reiss., *Recognizing Planar Objects Using Invariant Image Features, from Lecture notes in computer science*. Springer, 1993.

[17] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*, pp. 818–833, Springer, 2014.

[18] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Computer Vision and Pattern Recognition (CVPR)*, 2015.

[19] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[20] "Swedish leaf dataset." `http://www.cvl.isy.liu.se/en/research/datasets/swedish-leaf/`.

[21] O. J. O. Söderkvist, "Computer vision classification of leaves from swedish trees," 2001.

[22] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *in ICML Workshop on Deep Learning for Audio, Speech and Language Processing*, 2013.

[23] H. Kaiming, Z. Xiangyu, R. Shaoqing, and S. Jian, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification." `https://arxiv.org/abs/1502.01852`, 2015.

[24] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.

[25] "Code used in this paper." `https://github.com/dykuang/Leaf_Project`.

[26] "Uea & ucr time series classification repository." `http://timeseriesclassification.com/`.

[27] A. Bagnall, J. Lines, J. Hills, and A. Bostrom, "Time-series classification with cote: The collective of transformation-based ensembles.," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, pp. 2522–2535, 2015.

[28] L. van der Maaten and G. Hinton., "Visualizing high-dimensional data using t-sne.," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.

[29] "One-hundred plant species leaves data set data set." `https://archive.ics.uci.edu/ml/datasets/One-hundred+plant+species+leaves+data+set`.

[30] J. Lines, L. M. Davis, J. Hills, and A. Bagnall, "A shapelet transform for time series classification," in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '12, (New York, NY, USA), pp. 289–297, ACM, 2012.