Complementary Features Generation via Least-Squares Normal Transform (LNT): Methodology and Applications

Anonymous IEEE VCIP 2023 Submission

Abstract—**Although the idea of deriving new discriminant features from proper linear combinations of input features, denoted by x, is appealing, to determine the weights of linear combinations efficiently is an open question. The derived and original input features are called <mark>complementary and raw features, respectively.</mark> In this work, we present a novel supervised learning method to generate discriminant complementary features efficiently, named the least-squares normal transform (LNT). LNT consists of two steps. First, we convert a C-class classification problem to a binary classification problem. The two classes are assigned with 0 and 1, respectively. Next, we formulate a least-squares regression problem from the N-dimensional (N-D) feature space to the 1-D output space, and solve the least-squares normal equation to obtain one N-D normal vector, denoted by a_1. Since one normal vector is yielded by one binary split, we can obtain M normal vectors with M splits. Then, Ax is called an LNT of x, where transform matrix $A \in R^{M \times N}$ by stacking a_j^T, $j = 1, \cdots, M$, and the LNT, Ax, can generate M new features. The newly generated complementary features are shown to be more discriminant than the raw features. Experiments show that the classification performance can be improved by these new features.**

Index Terms—**green learning, supervised feature generation, image classification, least-sqaures normal equation, least-sqaures normal transform.**

I. INTRODUCTION

Deep learning (DL) has been dominating in the field of artificial intelligence (AI) and machine learning (ML) in the last decade. However, its black-box nature and heavy computational complexity are concerns to academia and industry. To build interpretable, reliable and sustainable learning systems, green learning (GL) has been proposed as an alternative in recent years [1]–[6]. GL consists of three modules: 1) unsupervised representation learning, 2) supervised feature learning, and 3) supervised decision learning. We focus on the second module in this work.

A rich set of representations is generated in the first module of GL without any supervision. They are rank-ordered according to their discriminant power by the discriminant feature test (DFT) [5] with supervision in the second module. Then, discriminant representations are chosen as features. Since there is no supervision adopted in Module 1, representations of GL may not be as competitive as those obtained by the end-to-end optimized DL approach. To address this shortcoming, an idea of creating new features by linearly combining selected features was investigated in [7], [8]. On the one hand, it was shown that it is feasible get more discriminant features through linear combinations. On the other hand, it was a nontrivial problem in finding the weights of good linear combinations. Several search algorithms were examined in [7], [8], including the probabilistic search, the adaptive particle swarm optimization (APSO) search, and the stochastic gradient descent (SGD) search. However, they are still computationally expensive.

The derived and original input features are called complementary and raw features, respectively. A novel and efficient supervised method to generate discriminant complementary features efficiently is proposed in this work. It is named the least-squares normal transform (LNT). LNT consists of two steps. First, a C-class classification problem is converted to a binary classification problem by merging individual classes into two major super-classes. They are assigned

with 0 and 1, respectively. Next, we formulate a least-squares regression problem from the N-dimensional (N-D) feature space to the 1-D output space, and solve the least-squares normal equation to obtain a N-D normal vector, denoted by a_1. Since one normal vector is yielded by one binary split, we have M normal vectors with M splits. Then, Ax is called an LNT of x, where transform matrix $A \in R^{M \times N}$ by stacking a_j^T, $j = 1, \cdots, M$, and the LNT, Ax, can generate M new features. The newly generated complementary features are shown to be more discriminant than the raw features. Experiments show that the classification performance can be improved by these new features.

Our work has two major contributions. First, a novel LNT is introduced as an efficient tool to generate more discriminant complementary features based on raw features. Second, the application of LNT to image classification problems is presented to demonstrate its power in boosting the classification performance in practice. The rest of this paper is organized as follows. The related background is reviewed in Sec. II. The LNT method and its application for new complementary feature generation are described in Sec. III. Experimental results with several classical image classification problems are shown in Sec. IV. Finally, the concluding remarks and future extensions are given in Sec. V.

II. BACKGROUND REVIEW

A. Image Classification with Neural Networks

Neural-network-based image classification methods have evolved for several decades, including early multi-layer perceptions (MLP) [9], [10], convolutional neural networks (e.g., LeNet [11] AlexNet [12], Resnet [13], Densenet [14], etc.), and the Vision Transformer (ViT) [15], [16]. Since the trend is to build deeper networks, it is called deep learning [17]. Neural-networks are end-to-end optimization systems. They have no clear boundary between feature extraction and decision making. Generally speaking, earlier and later layers play roles for feature extraction and decision making, respectively. Yet, research was conducted [18] to shed light on the role of deep features.

B. Green Learning

One concern with deep learning is the high carbon footprint. GL has been developed to address this issue. GL systems are characterized by smaller model sizes, lower computational complexity in both training and inference stages (thus, a lower carbon footprint). GL has been applied to a range of applications with competitive performance. Successful GL's application examples include: image classification [19]–[21], texture and image generation [22]–[24], low-resolution face recognition [25], face gender classification [26], deepfake detection [27], [28], blind image quality assessment [29], anomaly detection [30], image forensics [31], [32], disease classification [33], point cloud classification, segmentation and registration [34]–[38]. GL can be conveniently deployed on mobile and edge devices. It can also save a significant amount of energy consumption at cloud centers. GL consists of three modules: 1) unsupervised representation

learning, 2) supervised feature learning, and 3) supervised decision learning. Although our current work focus on the second module, it also involves the first module. Both modules are briefly reviewed below.

1) Unsupervised Representation Learning: One example of unsupervised representation learning is the histogram of oriented gradients (HoG) [39], [40]. The HoG features were popular in computer vision before the deep learning era. Their main limitation is that they do not capture long range dependency effectively. The unsupervised representation learning of GL is built upon upon two ingredients: 1) the Saab transform [4] and 2) successive subspace learning (SSL) [19] via the cascade of channel-wise (C/w) Saab transforms [20] in multiple stages/scales. The Saab transform is a variant of principal component analysis (PCA). It computes the mean of pixel values of a local patch and treats it as the DC (direct-current) component. The DC-removed pixel values of local patches are zero-mean random vectors. Thus, PCA can be applied, yielding multiple AC (alternating current) components. They correspond to the DC and AC Saab coefficients, respectively. The DC and low-frequency AC coefficients can be down-sampled to a coarser grid, known as the pooling operation. The max-pooling is used in DL while the absolute max-pooling is more effective in GL [40]. The Saab transform is a data-driven transform. It is neither hand-crafted by humans nor derived from labels through back propagation. The multi-stage Saab transforms are conducted in a one-pass feedforward manner.

2) Supervised Feature Learning: A rich set of representations is created in the first module. They are rank-ordered according to their discriminant power by the discriminant feature test (DFT) [5] with supervision in the second module. If the representation set does not contain highly discriminant ones, DFT cannot generate new features. One way to derive new discriminant features is to conduct linear combinations of existing features. It is an open question to determine the weights efficiently. In the next section, we present a supervised method to yield new discriminant features.

III. PROPOSED LNT METHOD

The LNT method in the context of image classification is proposed in this section. First, we describe a pre-processing step that converts a classification problem to a regression problem in Sec. III-A. Next, the LNT algorithm is presented in Sec. III-B. Finally, to rank the discriminant power of generated complementary features, a post-processing step is discussed in Sec. III-C.

A. Pre-processing: Mapping from Class Labels to Binary Values

We can build a link between the multi-object classification problem and the regression problem as follows. For C object classes, unit vectors in the R^C space define C one-hot vectors, each of which is used to denote an object class. If each input image has N features, the classification problem can be formulated as a nonlinear regression problem in form of $f : R^N \to R^C$. Although a nonlinear regression function is needed for higher classification accuracy, we consider its linear approximation here for two reasons. First, we are only interested in generating new features here and will leave the nonlinear operation to the tree-based classifier (e.g., the XGBoost classifier). Second, it is desired to have an efficient computational algorithm. The linear regression problem is well studied and allows fast computation.

We can rewrite $f : R^N \to R^C$ as

$$A\mathbf{x}_j + \mathbf{b} \approx \mathbf{w}_j, \quad j = 1, \cdots, C, \tag{1}$$

where $A \in R^{C \times N}$ is a matrix, $\mathbf{b} \in R^C$ is a bias vector, $\mathbf{x}_j \in R^N$ and $\mathbf{w}_j \in R^C$ denote a feature vector of an image input belonging

to the jth class and the one-hot vector associated with the jth class, respectively. Furthermore, if $\mathbf{a}_i^T$ is the ith row of matrix A, we have

$$\mathbf{a}_i^T \mathbf{x}_j + \mathbf{b}_i \approx \delta_{i,j}, \quad i, j = 1, \cdots, C, \tag{2}$$

where $\delta_{i,j}$ is the Kronecker delta. It is equal to one if $i = j$. Otherwise, it is equal to zero. Clearly, $\mathbf{a}_i^T \in R^N$ is the weight vector of a linear combination of input features. The Kronecker delta in Eq. (2) can be interpreted as a one-versus-others split. If the number, L, of training samples is greater than N, $\mathbf{a}_i$ can be solved using the linear least-squares regression as presented in Sec. III-B.

To enrich the set of weight vectors, we adopt various grouping schemes, such as "1 vs. (C-1)", "2 vs. (C-2)", $\cdots$, "k vs. (C-k)", where $k = \lfloor C/2 \rfloor$. For a specific "l vs. (C-l)" grouping, we have $\binom{C}{l}$ splits. As a result, the total number of splits can be computed via

$$M = \sum_{l=1}^{k} \binom{C}{l}, \qquad k = \lfloor C/2 \rfloor. \tag{3}$$

To give an example, we have $C = 10$ and $M = 637$ for several commonly used datasets such as MNIST and CIFAR-10. One split partitions C classes into two super-classes with labels "0" and "1".

B. Least-Squares Normal Transform (LNT)

We associate each split with a target vector $\mathbf{t} \in R^C$, whose elements are either zero or one. They serve as indicators for one of the two super classes. With L training samples, we can formulate the following linear least-squares regression problem to solve for M weight vectors simultaneously:

$$AX + B = T, \tag{4}$$

where $A \in R^{M \times N}$ is the weight matrix, $X \in R^{N \times L}$ is the data matrix, $B \in R^{M \times L}$ is the bias matrix, and $T \in R^{M \times L}$ is the indicator matrix. Since all training samples share the same bias vector, B is a rank-one matrix with the same column vector repeated L times. The element $t_{m,l}$ in T takes values 0 or 1 depending on split scheme $1 \le m \le M$ and the super-class label of the lth training sample.

By taking the expectation of Eq. (4), we have

$$B = E[T] - AE[X]. \tag{5}$$

Actually, we are not concerned with B but A since we are only interested in generating new complementary features. Since the bias term shifts a feature with a certain value, it has no effect on its discriminant power. Here, we focus on the solution of weight matrix A only. Matrix A can be obtained by solving the least-squares normal equation. It has a closed form solution in form of

$$A = TX^T (XX^T)^{-1}. \tag{6}$$

For a training or testing sample with its raw feature vector $\mathbf{x}$, its M complementary features can be generated via

$$A\mathbf{x} = \mathbf{p}, \tag{7}$$

where matrix A is obtained from Eq. (6) and $\mathbf{p} = (p_1, \cdots, p_M)$, and where p_m is the value of the mth generated feature. Eq. (7) is called the least-squares normal transform (LNT).

C. Post-processing: Selection of Discriminant Generated Features

After feature generation, we apply the discriminant feature test (DFT) [5] to complementary features to find their discriminant power. DFT is a semi-supervised feature selection method. It partitions the dynamic range of a 1-D feature into two non-overlapping intervals and computes the weighted entropy loss from them. Furthermore,
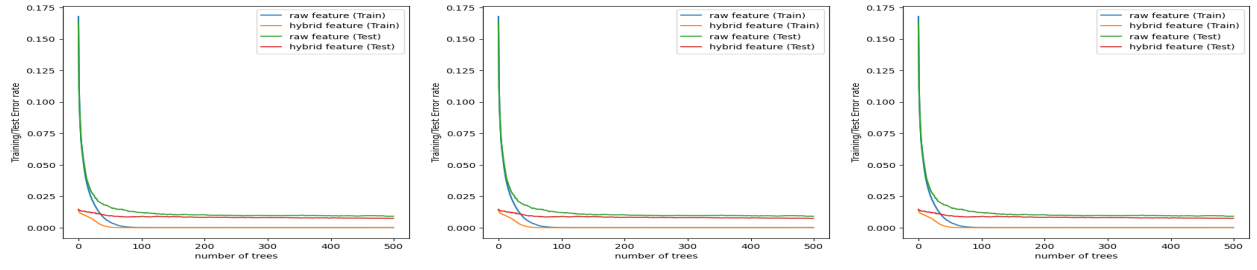
Fig. 1. Comparison of the discriminant power of raw and generated features of three datasets (from left to right): MNIST, Fashion MNIST, and CIFAR-10. The DFT loss values of raw and generated features are shown in blue and red points in the curve, respectively. A feature is more discriminant if its DFT loss value is smaller.
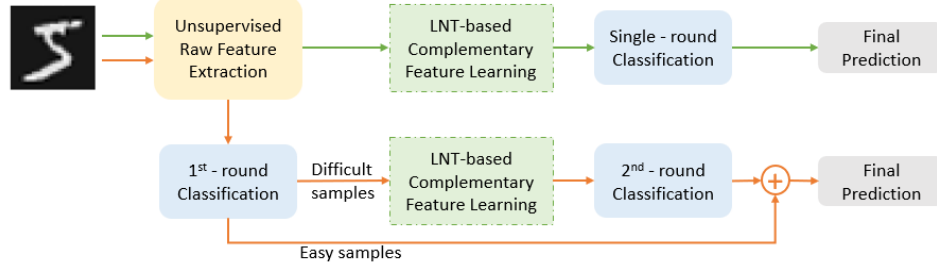

Fig. 2. The block diagram of the image classification system, which follows the standard GL data processing pipeline. We propose one-round of two-round decision systems and indicate them in green and red arrows, respectively.

it searches for the minimum weighted entropy loss among a set of uniformly sampled points and uses it as the DFT loss of the feature. A feature is more discriminant if its DFT loss value is smaller. One sort all features according to their DFT loss values from the smallest to the largest to obtain a DFT loss curve (see Fig. 1) and use the elbow point of the DFT curve to select a set of discriminant features for decision making. We compare the discriminant power of raw and newly generated complementary features for the MNIST, Fashion-MNIST, and CIFAR-10 three datasets in Fig. 1. The DFT loss values of raw and generated features are shown in blue and red points in the curve, respectively. We see clearly that the generated complementary features have higher discriminability than the raw features.

IV. EXPERIMENTS

We demonstrate the performance improvement of image classification systems by using the proposed LNT method in this section.

A. Experimental Setup

TABLE I
THE HYPER-PARAMETERS OF THE PIXELHOP++ ARCHITECTURE FOR
MNIST, F-MNIST AND CIFAR-10.

	Filter Spatial Size (Channel #)		
	MNIST	F-Mnist	CIFAR-10
Hop-1	5x5 (25)	5x5 (25)	5x5 (69)
Hop-2	5x5 (256)	5x5 (210)	5x5 (576)
Hop-3	5x5 (652)	5x5 (469)	5x5 (1441)

We conduct experiments on MNIST [41], Fashion-Mnist (F-Mnist) [42], and CIFAR-10 [43] three datasets. The image classification system is shown in Fig. 2. It follows the standard GL data processing pipeline. A 3-hop pixelhop++ [20] architecture is used for unsupervised raw feature extraction. Its hyper-parameters are given in Table

I. Only the Saab coefficients in the last hop are used as raw features. For color images, we covert the RGB channels to YUV channels, apply the same 3-hop pixelhop++ to each channel, and use all of them as raw features. Furthermore, the HOG features are included in the raw feature set for F-Mnist and CIFAR-10 to increase feature diversity.

In the one-round system, we apply LNT to the original feature set directly to generate complementary features, and train an XGBoost classier with DFT selected features to predict the image class. In the two-round system, we use the raw feature set for decision making in the first round. Then, we categorize samples into easy and difficult ones based on their confidence scores. The confidence score is calculated as the entropy of the predicted label vector. A higher entropy implies a less confident decision. The round-one decision of an easy test sample is accepted as is. On the other hand, a difficult test sample will be fed into the second-round decision pipeline, where LNT is used to generate complementary features and another XGBoost classifier is trained. The prediction in the second round is accepted as the prediction of difficult samples.

In the one-round system, all samples go through the LNT process so that its computational complexity in the inference stage is higher. In the two-round system, we can lower the inference complexity since easy samples do not go through the LNT process. On the other hand, the two-round system has a larger model size (i.e., more model parameters) than the one-round system. Thus, the one- and two-round designs provide a tradeoff between computational and storage complexities. Furthermore, for the performance benchmarking purpose, the one-round system with raw features only (i.e., without LNT generated features) is called the baseline method.

B. Performance Evaluation

We compare the classification performance of six methods; namely, LeNet-5 [11], PixelHop [19], PixelHop$^+$ [19], the baseline, one-round and two-round classification systems proposed in Table II. In
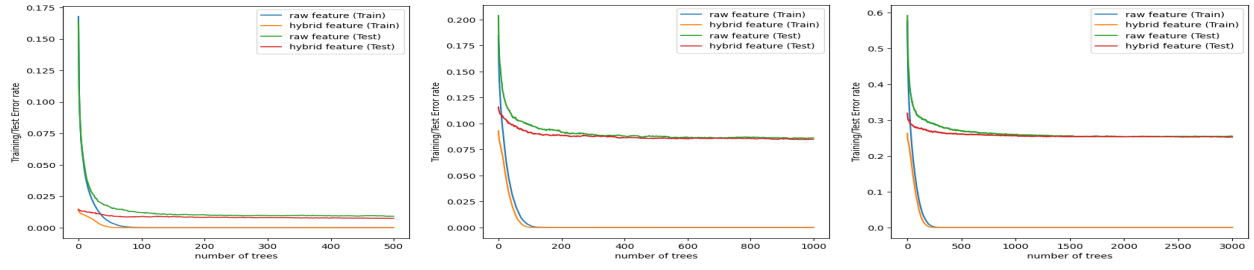
Fig. 3. The error rate as a function of the tree number in baseline and one-round system for MNIST (left), F-Mnist (middle) and CIFAR-10 (right). For comparison, we set the same tree depth for the baseline and the one-round system.

TABLE II
COMPARISON OF CLASSIFICATION ACCURACY ON MNIST, F-MNIST, CIFAR-10 WITH LENET-5, PIXELHOP, PIXELHOP$^+$, THE PROPOSED ONE-ROUND AND TWO-ROUND SYSTEMS.

	Test Accuracy (%)		
	MNIST	F-Mnist	CIFAR-10
LeNet-5	99.07	89.54	68.72
PixelHop	98.90	91.30	71.37
PixelHop$^+$	99.09	91.68	72.66
Baseline (Ours)	99.09	91.38	74.51
1-round system (Ours)	99.24	91.75	74.80
2-round system (Ours)	**99.32**	**92.07**	**75.93**

TABLE III
HYPER PARAMETERS AND MODEL SIZES OF THE XGBOOST CLASSIFIERS.

Dataset	Setting	depth	tree#	Model Size
MNIST	baseline	3	500	47k
MNIST	1-round	2	500	23k
MNIST	2-round	2	500+200	32.2k
F-Mnist	baseline	5	1000	382k
F-Mnist	1-round	3	2000	188k
F-Mnist	2-round	4	500+1200	323k
CIFAR-10	baseline	4	3000	570k
CIFAR-10	1-round	3	3000	282k
CIFAR-10	2-round	4	1500+1500	570k

the one-round system, all training samples are used to obtain the LNT parameters. In the two-round system, we pick c_1% training samples and c_2% testing samples to go to the second round, where $c_1 = 20, 60, 80$ and $c_2 = 2, 20, 20$ for MNIST, F-Mnist, CIFAR-10, respectively. In addition, we select 200 most discriminant complementary features for MNIST and use all 637 complementary features in F-Mnist and CIFAR-10. We take the union of raw features and selected complementary features and feed them into the XGBoost

TABLE IV
COMPARISON OF COMPUTATIONAL COMPLEXITY IN THE INFERENCE STAGE IN TERMS OF THE AVERAGED FLOATING POINT OPERATIONS (FLOPs) REQUIRED TO CLASSIFY AN IMAGE, WHERE ONLY COMPUTATIONS IN THE COMPLEMENTARY FEATURE GENERATION AND THE XGBOOST CLASSIFICATION ARE CONSIDERED.

	FLOPs #		
	MNIST	F-Mnist	CIFAR-10
Baseline system	0.657M	2.686M	14.799M
1-round system	3.039M	13.767M	30.039M
2-round system	0.491M	3.949M	16.500M

classifier in the final stage. The hyper parameters of the XGBoost classifiers used in the baseline, one-round and two-round systems are summarized in Table III.

a) Comparison of Correct Classification Rates. We see from Table II that the introduction of LNT-generated complementary features boost the baseline performance to a higher level. The 2-round system outperforms the baseline by 0.23%, 0.67% and 1.42% in MNIST, F-Mnist, and CIFAR-1, respectively. The one-round system also outperforms the baseline in all datasets. Furthermore, complementary features help the XGBoost classifier to converge faster as shown in Fig. 3. Let us focus on the green and red curves in all three subfigures, which show the convergence curves in the inference stage of the baseline and the round-one system, respectively. Clearly, the one-round system converges faster than the baseline in all datasets.

b) Comparison of Model Sizes. We compare the model sizes of the XGBoost classifiers of the baseline, one-round and two-round systems in Table III. The model size of the one-round system is around 50% of that of the baseline. Although there are two XGBoost classifiers in the tw-round system, its total model size is still slightly lower than that of the baseline. The advantages of introducing complementary features are demonstrated by higher classification accuracy and smaller model sizes.

c) Comparison of Computational Complexity in Inference. We compare the computational complexity in the inference stage in terms of the averaged floating point operations (FLOPs) required to classify one image in Table IV. Since these systems share the same raw feature generation module, we only compare the number of FLOPs required by the complementary feature generation and the XGBoost classification. Since each image has $32 \times 32 = 1,024$ pixels. The complexity can be normalized per pixel by dividing the numbers in Table IV with 1024. We see that the two-round system has much lower FLOPs than the one-round system, which is only 16.1%, 28.6%, 54.92% of the one-round system for MNIST, F-Mnist, CIFAR-10, respectively. The computational cost of the two-round system is comparable with that of the baseline.

V. CONCLUSION AND FUTURE WORK

A novel supervised complementary feature generation method based on the least-squares normal transform (LNT), was proposed in this work. The generated complementary features are more discriminant than raw features. The use of newly generated complementary features in the image classification task was demonstrated. It was shown experimentally that they boost the classification performance and improves the convergence behavior of the XGBoost classifer. As future extensions, we would like shed more light on LNT generated features and explore their applications to a wider range of image processing and computer vision tasks.

REFERENCES

[1] C.-C. J. Kuo, "Understanding convolutional neural networks with a mathematical model," *Journal of Visual Communication and Image Representation*, vol. 41, pp. 406–413, 2016.

[2] ——, "The CNN as a guided multilayer RECOS transform," *IEEE signal processing magazine*, vol. 34, no. 3, pp. 81–89, 2017.

[3] C.-C. J. Kuo and Y. Chen, "On data-driven Saak transform," *Journal of Visual Communication and Image Representation*, vol. 50, pp. 237–246, 2018.

[4] C.-C. J. Kuo, M. Zhang, S. Li, J. Duan, and Y. Chen, "Interpretable convolutional neural networks via feedforward design," *Journal of Visual Communication and Image Representation*, vol. 60, pp. 346–359, 2019.

[5] Y. Yang, W. Wang, H. Fu, and C.-C. J. Kuo, "On supervised feature selection from high dimensional feature spaces," *arXiv preprint arXiv:2203.11924*, 2022.

[6] C.-C. J. Kuo and A. M. Madni, "Green learning: Introduction, examples and outlook," *Journal of Visual Communication and Image Representation*, vol. 90, p. 103685, 2023.

[7] H. Fu, Y. Yang, V. K. Mishra, and C.-C. J. Kuo, "Subspace learning machine (slm): Methodology and performance," in *2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[8] H. Fu, Y. Yang, Y. Liu, J. Lin, E. Harrison, V. K. Mishra, and C.-C. J. Kuo, "Acceleration of subspace learning machine via particle swarm optimization and parallel processing," in *2022 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 2022, pp. 1019–1024.

[9] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain." *Psychological review*, vol. 65, no. 6, p. 386, 1958.

[10] R. Lin, Z. Zhou, S. You, R. Rao, and C.-C. J. Kuo, "From two-class linear discriminant analysis to interpretable multilayer perceptron design," *arXiv preprint arXiv:2009.04442*, 2020.

[11] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[14] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017.

[16] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[17] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–44, 05 2015.

[18] H. Xu, Y. Chen, R. Lin, and C.-C. J. Kuo, "Understanding cnn via deep features analysis," in *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 2017, pp. 1052–1060.

[19] Y. Chen and C.-C. J. Kuo, "Pixelhop: A successive subspace learning (ssl) method for object recognition," *Journal of Visual Communication and Image Representation*, vol. 70, p. 102749, 2020.

[20] Y. Chen, M. Rouhsedaghat, S. You, R. Rao, and C.-C. J. Kuo, "Pixelhop++: A small successive-subspace-learning-based (ssl-based) model for image classification," in *2020 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2020, pp. 3294–3298.

[21] Y. Yang, V. Magoulianitis, and C.-C. J. Kuo, "E-pixelhop: An enhanced pixelhop method for object classification," in *2021 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2021, pp. 1475–1482.

[22] X. Lei, G. Zhao, K. Zhang, and C.-C. J. Kuo, "Tghop: an explainable, efficient, and lightweight method for texture generation," *APSIPA Transactions on Signal and Information Processing*, vol. 10, p. e17, 2021.

[23] X. Lei, W. Wang, and C.-C. J. Kuo, "Genhop: An image generation method based on successive subspace learning," in *2022 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2022, pp. 3314–3318.

[24] Z. Azizi, C.-C. J. Kuo *et al.*, "Pager: Progressive attribute-guided extendable robust image generation," *APSIPA Transactions on Signal and Information Processing*, vol. 11, no. 1, 2022.

[25] M. Rouhsedaghat, Y. Wang, S. Hu, S. You, and C.-C. J. Kuo, "Low-resolution face recognition in resource-constrained environments," *Pattern Recognition Letters*, vol. 149, pp. 193–199, 2021.

[26] M. Rouhsedaghat, Y. Wang, X. Ge, S. Hu, S. You, and C.-C. J. Kuo, "Facehop: A light-weight low-resolution face gender classification method," in *International Conference on Pattern Recognition*. Springer, 2021, pp. 169–183.

[27] H.-S. Chen, M. Rouhsedaghat, H. Ghani, S. Hu, S. You, and C.-C. J. Kuo, "Defakehop: A light-weight high-performance deepfake detector," in *2021 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2021, pp. 1–6.

[28] H.-S. Chen, S. Hu, S. You, C.-C. J. Kuo *et al.*, "Defakehop++: An enhanced lightweight deepfake detector," *APSIPA Transactions on Signal and Information Processing*, vol. 11, no. 2, 2022.

[29] Z. Mei, Y.-C. Wang, X. He, and C.-C. J. Kuo, "Greenbiqa: A lightweight blind image quality assessment method," in *2022 IEEE 24th International Workshop on Multimedia Signal Processing (MMSP)*. IEEE, 2022, pp. 1–6.

[30] K. Zhang, B. Wang, W. Wang, F. Sohrab, M. Gabbouj, and C.-C. J. Kuo, "Anomalyhop: an ssl-based image anomaly localization method," in *2021 International Conference on Visual Communications and Image Processing (VCIP)*. IEEE, 2021, pp. 1–5.

[31] Y. Zhu, X. Wang, H.-S. Chen, R. Salloum, and C.-C. J. Kuo, "A-pixelhop: A green, robust and explainable fake-image detector," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 8947–8951.

[32] Y. Zhu, X. Wang, R. Salloum, H.-S. Chen, and C.-C. Kuo, "Rggid: A robust and green gan-fake image detector," *APSIPA Transactions on Signal and Information Processing*, vol. 11, 01 2022.

[33] X. Liu, F. Xing, C. Yang, C.-C. J. Kuo, S. Babu, G. El Fakhri, T. Jenkins, and J. Woo, "Voxelhop: Successive subspace learning for als disease classification using structural mri," *IEEE journal of biomedical and health informatics*, vol. 26, no. 3, pp. 1128–1139, 2021.

[34] M. Zhang, H. You, P. Kadam, S. Liu, and C.-C. J. Kuo, "Pointhop: An explainable machine learning method for point cloud classification," *IEEE Transactions on Multimedia*, 2020.

[35] P. Kadam, M. Zhang, S. Liu, and C.-C. J. Kuo, "R-pointhop: A green, accurate, and unsupervised point cloud registration method," *IEEE Transactions on Image Processing*, vol. 31, pp. 2710–2725, 2022.

[36] M. Zhang, P. Kadam, S. Liu, and C.-C. J. Kuo, "Gsip: Green semantic segmentation of large-scale indoor point clouds," *Pattern Recognition Letters*, vol. 164, pp. 9–15, 2022.

[37] P. Kadam, M. Zhang, J. Gu, S. Liu, and C.-C. J. Kuo, "GreenPCO: An unsupervised lightweight point cloud odometry method," in *2022 IEEE 24th International Workshop on Multimedia Signal Processing (MMSP)*. IEEE, 2022, pp. 01–06.

[38] P. Kadam, H. Prajapati, M. Zhang, J. Xue, S. Liu, and C.-C. J. Kuo, "S3I-PointHop: SO(3)-invariant pointhop for 3d point cloud classification," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[39] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, 2005, pp. 886–893 vol. 1.

[40] Y. Yang, H. Fu, and C.-C. J. Kuo, "Design of supervision-scalable learning systems: Methodology and performance benchmarking," *arXiv preprint arXiv:2206.09061*, 2022.

[41] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[42] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.

[43] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," University of Toronto, Toronto, Ontario, Tech. Rep., 2009.