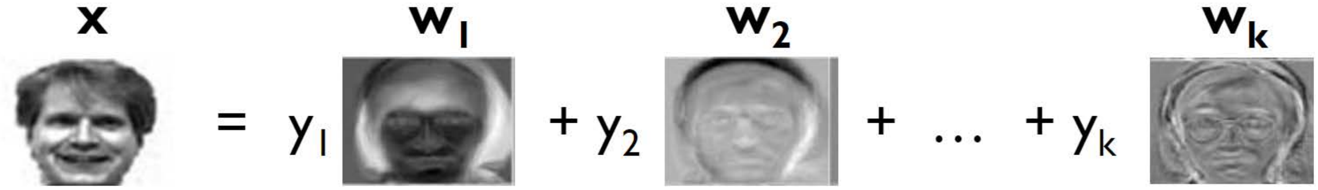
Image Classification with PixelHop and PixelHop++

C.-C. Jay Kuo
University of Southern California

Subspace Learning

❖ Unsupervised subspace learning

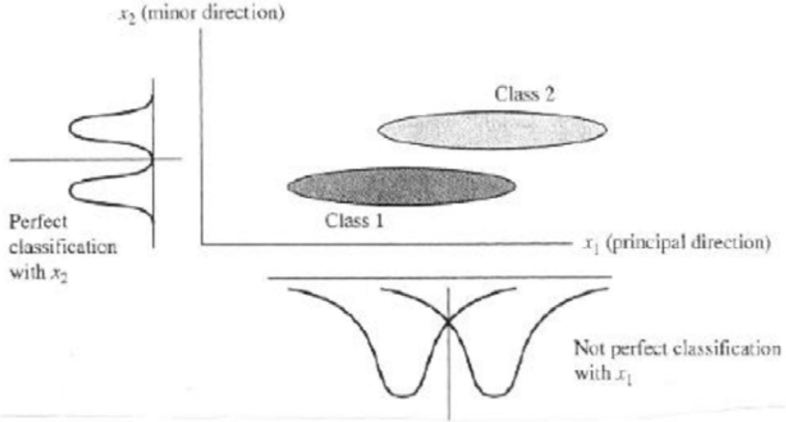
- Explore statistical properties of underlying signals
- Find a dominant and less redundancy feature space
- Methods:
  - Principal Component Analysis (PCA),
  - PCANet, Saaktransform, Saab transform



EigenFaces [1]

Subspace Learning

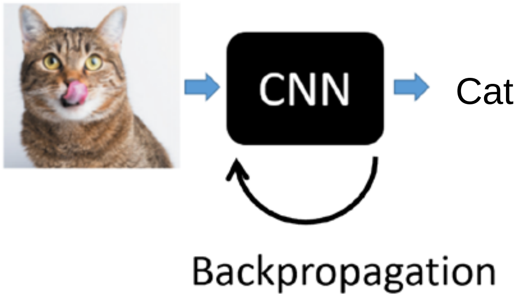
- ❖ Supervised subspace learning
  - Explore label information
  - Each class represented by its own subspace
  - Methods:
    - Supervised PCA
    - Linear Discriminant Analysis (LDA)



PCA vs LDA

Successive Subspace Learning

- ❖ Handle the weaknesses of CNN-based methods
  - ➢ Efficiency
  - ➢ Scalability
  - ➢ Explainable
- ❖ New Machine Learning Paradigm
  - ➢ Without backpropagation (BP)
  - ➢ Mathematically explainable and scalable

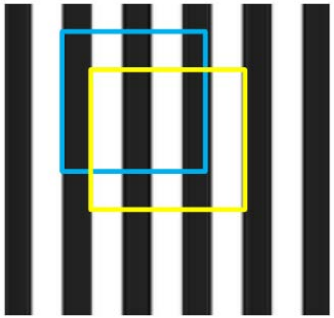


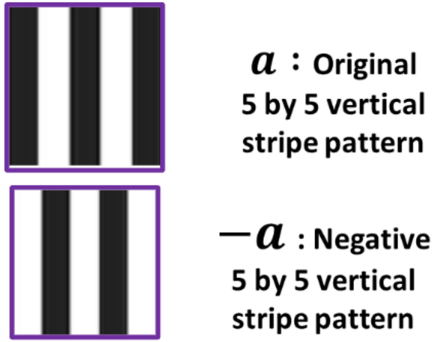
Saab Transform

- Saab (Subspace approximation with adjusted bias) transform
- Separate the transform kernels (anchor vector) to two categories
  - DC kernel: $\mathbf{a}_0 = \frac{1}{\sqrt{N}}(1, 1, \cdots, 1)^T$
  - AC kernel: PCA basis vectors
- Bias selection:
  - Two constraints:
    - Non-negative response constraint
    - Constant bias constraint

Saab Transform

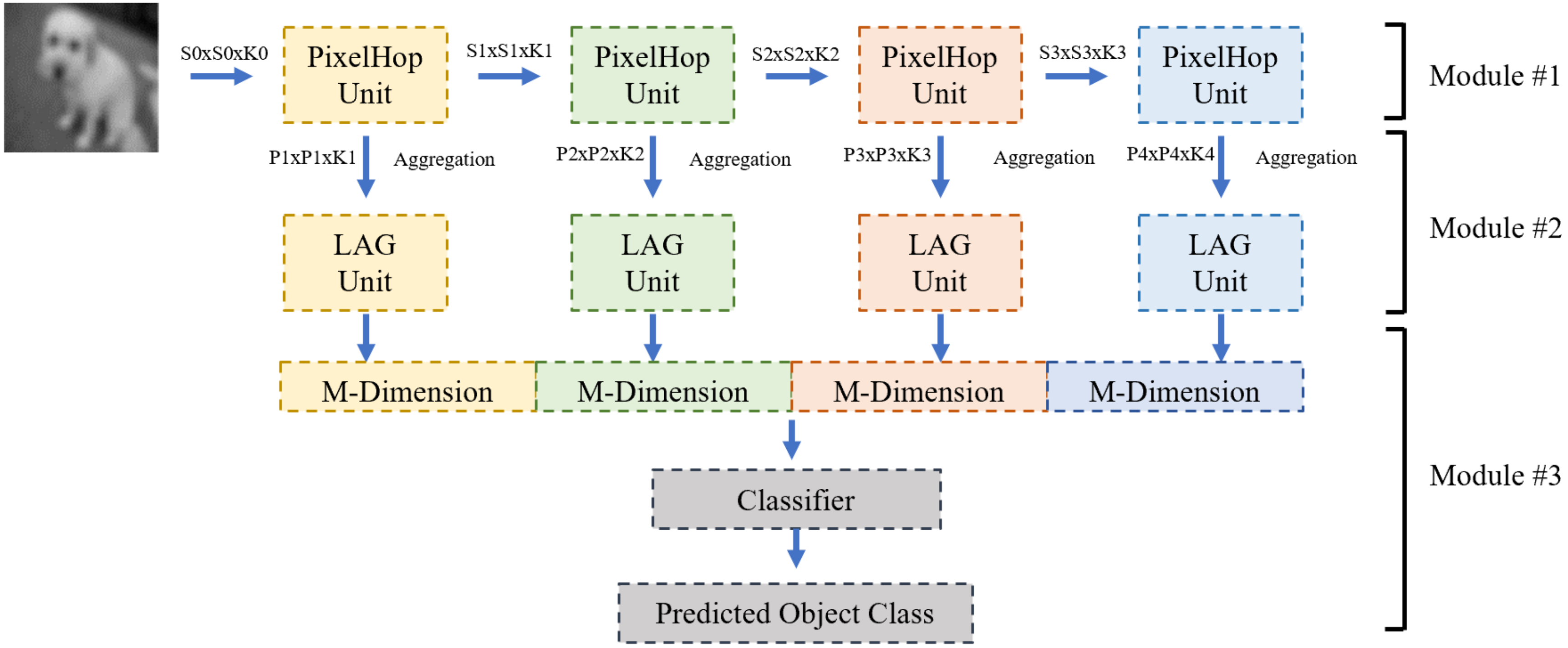
- ➢ Eliminate the sign confusion problem
  - ○ Confusing Case #1
    - ■ A positive response followed by a positive outgoing kernel weight
    - ■ A negative response followed by a negative outgoing kernel weight
  - ○ Confusing Case #2
    - ■ A positive response followed by a negative outgoing kernel weight
    - ■ A negative response followed by a positive outgoing kernel weight





PixelHop Method

System Overview



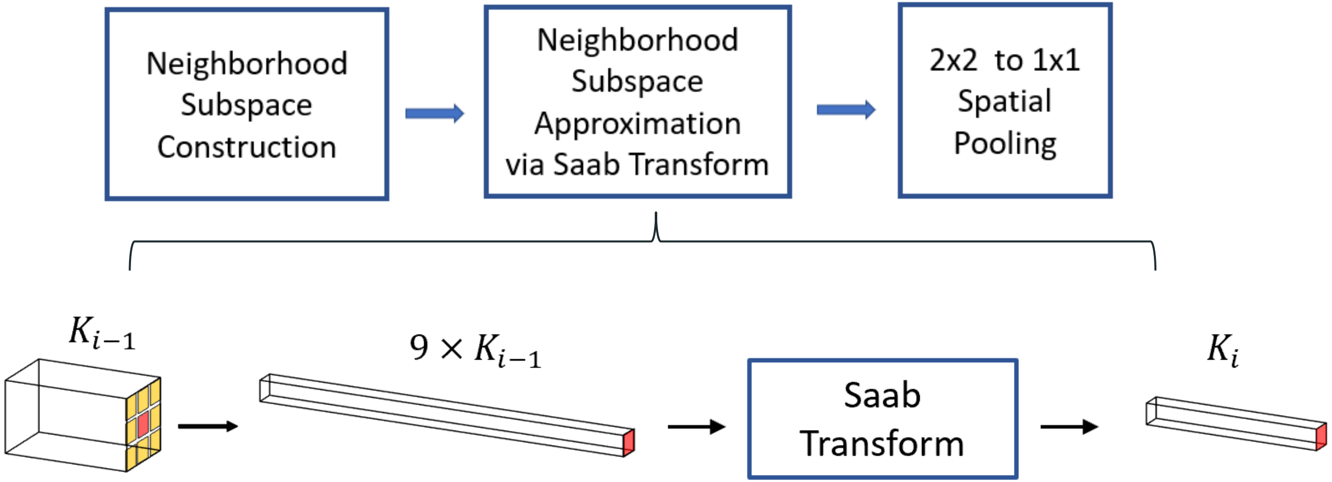
PixelHop Method

❖ Module #1: A Sequence of PixelHop Units in Cascade

- ➢ Derive attributes from near-to-far neighborhoods
- ➢ Apply a spatial max-pooling

PixelHop Method

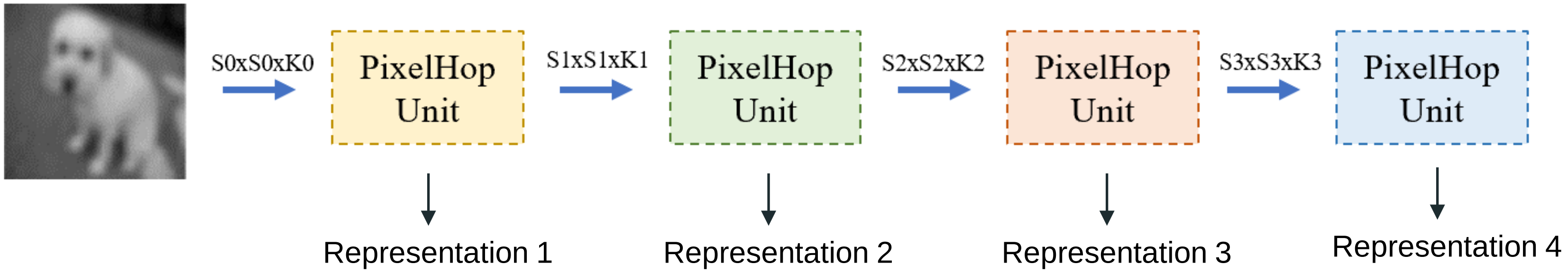
❖ PixelHop Unit



PixelHop Method

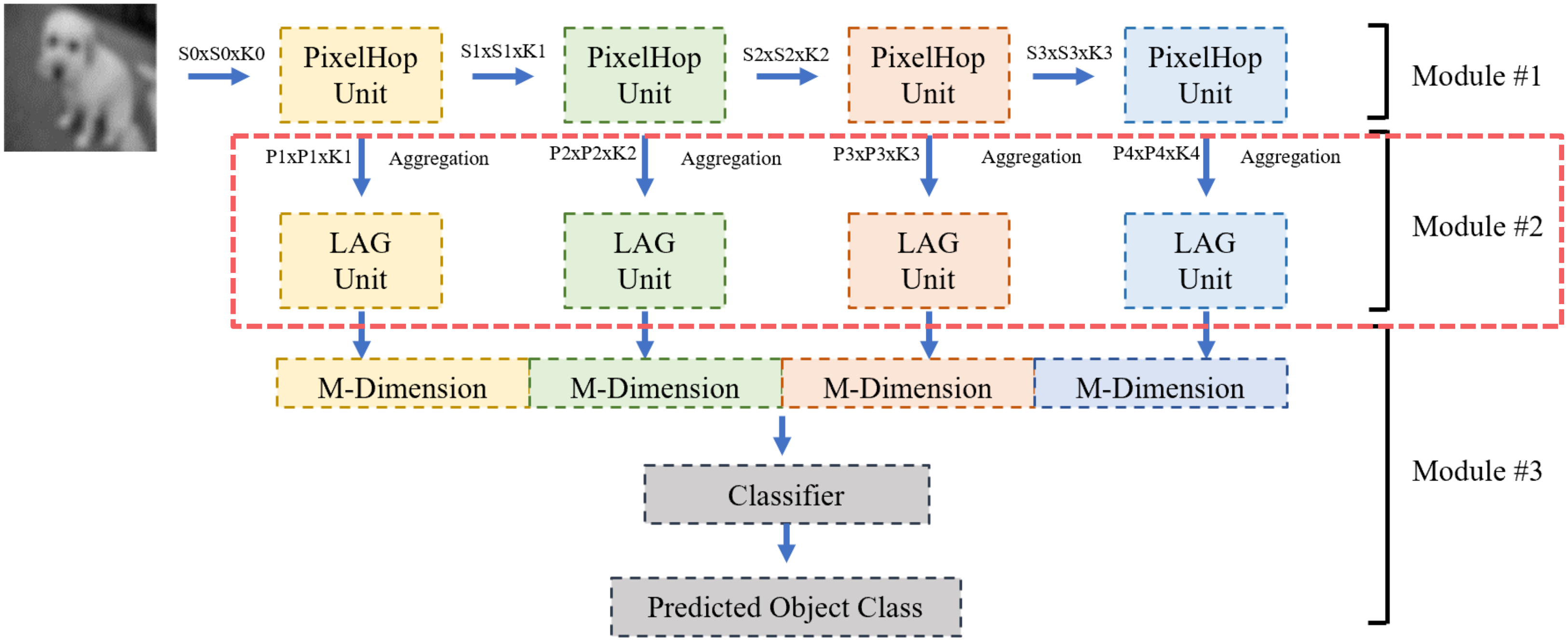
❖ PixelHop Unit

- Generate a set of new image representations
- Control the dimension of new subspace: Saab transforms
- Enlarge the neighboring size: spatial pooling



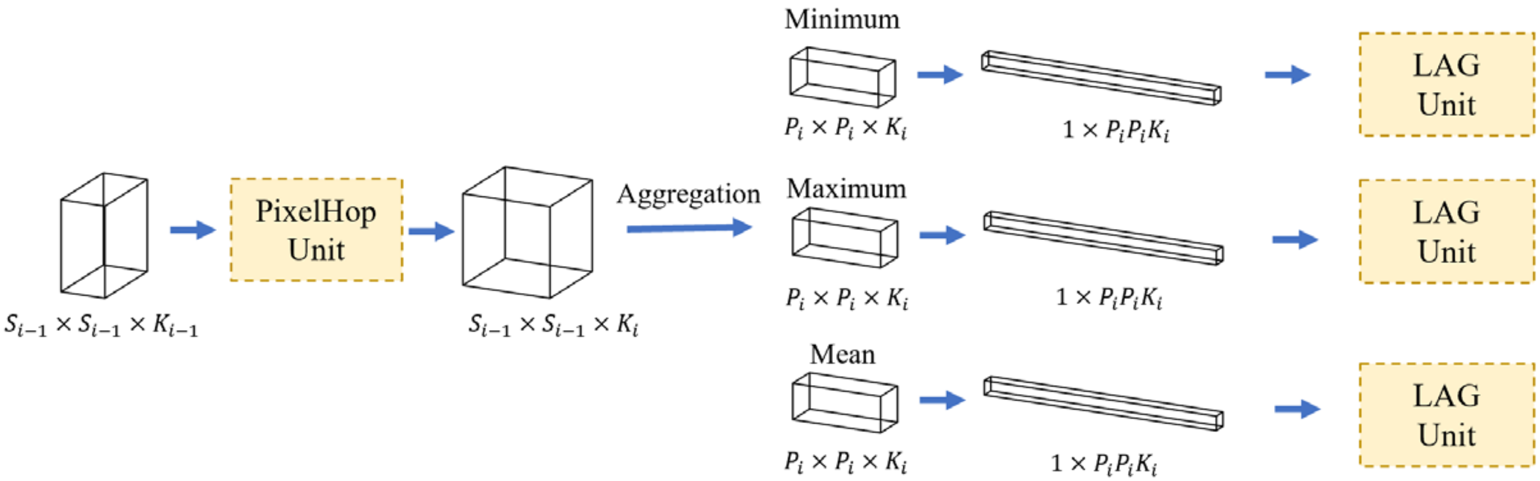
PixelHop Method

❖ Module #2: Aggregation and Supervised Subspace Learning



PixelHop Method

- ❖ Module #2: Aggregation and Supervised Subspace Learning
  - ➢ Aggregation: minimum, maximum, mean values



PixelHop Method

- Module #2: Aggregation and Supervised Subspace Learning
  - Supervised Subspace Learning : further reduce feature dimensions
  - Procedures of Label-Assisted reGression (LAG):
    - Learn subspace formed by samples of the same class
    - Set up a least-squared regression (LSR) system
    - Apply the learned regression matrix

PixelHop Method

❖ Label-Assisted Regression

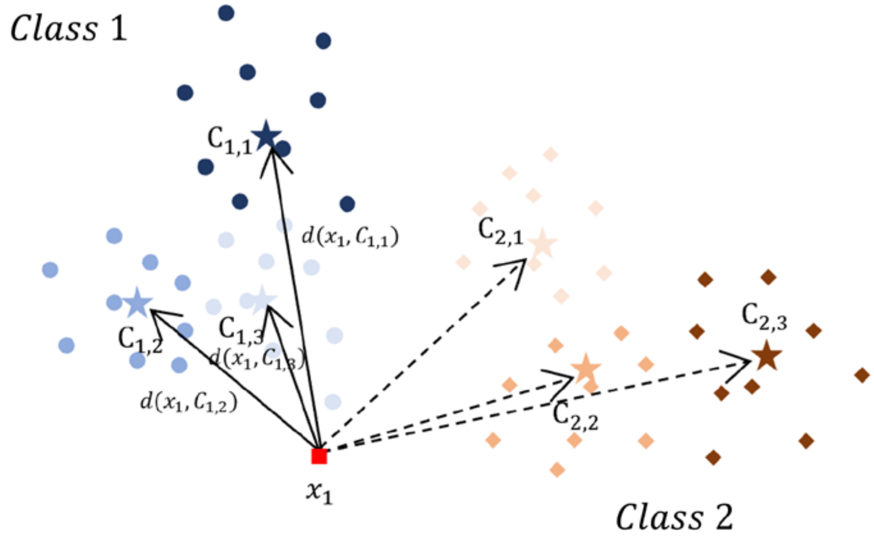
- Cluster samples of the same class to create object-oriented subspaces

- A sample from class j:

$$\mathbf{x}_j = (x_{j,1}, x_{j,2}, \cdots, x_{j,n})^T \in R^n$$

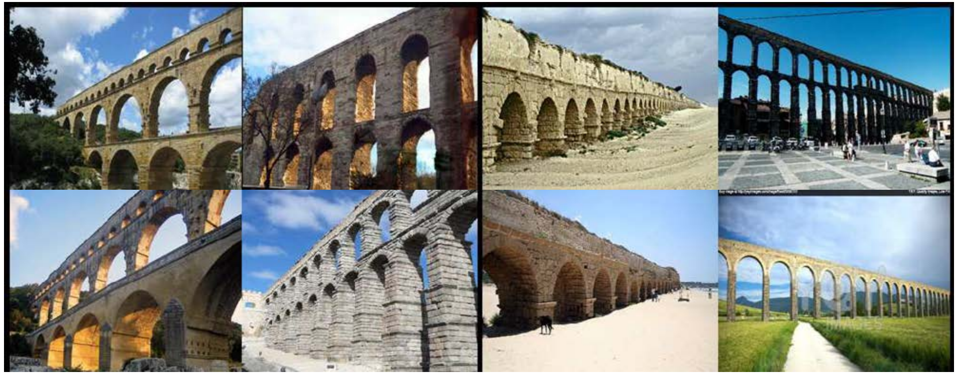
- Cluster centroids from class j:

$$\mathbf{c}_{j,1},\ \mathbf{c}_{j,2},\ \cdots,\ \mathbf{c}_{j,L}$$



PixelHop Method

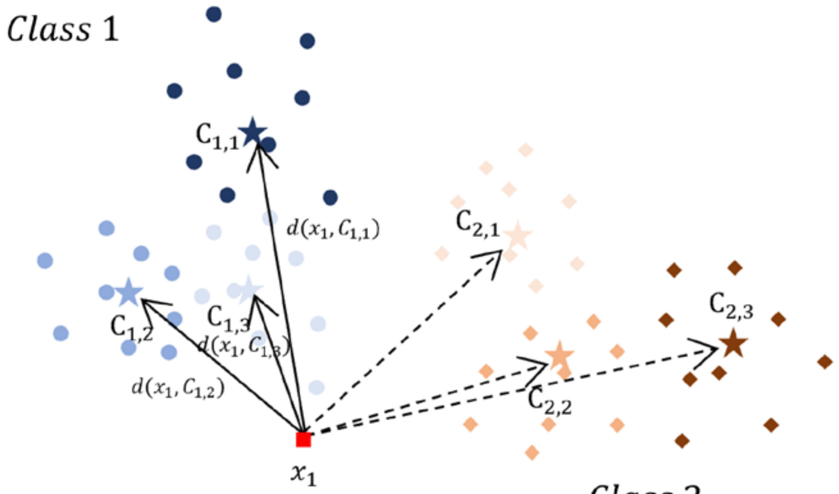
- ❖ Label-Assisted Regression
    - ➢ Representative samples to capture the diversity of a single class



PixelHop Method

❖ Label-Assisted Regression

- Compute a probability vector based on Euclidean distance



$$\mathrm{Prob}(\mathbf{x}_j, \mathbf{c}_{j',l}) = 0 \quad \text{if } j \neq j', \quad \mathrm{Prob}(\mathbf{x}_j, \mathbf{c}_{j,l}) = \frac{\exp(-\alpha d(\mathbf{x}_j, \mathbf{c}_{j,l}))}{\sum_{l=1}^{L} \exp(-\alpha d(\mathbf{x}_j, \mathbf{c}_{j,l}))},$$

- Target output vector:

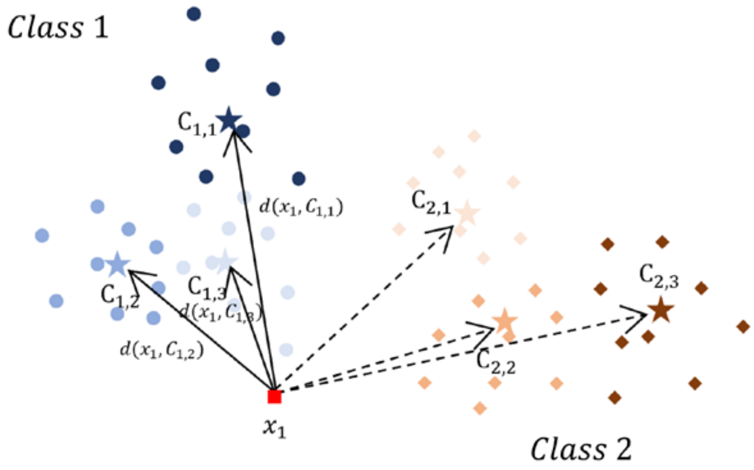
$$\mathrm{Target}(x_1) = \begin{bmatrix} P_1(x_1) \\ P_2(x_1) \end{bmatrix} = \begin{bmatrix} Prob(x_1, c_{1,1}) \\ Prob(x_1, c_{1,2}) \\ Prob(x_1, c_{1,3}) \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

PixelHop Method

❖ Label-Assisted Regression

➢ Set up and solve a linear LSR problem

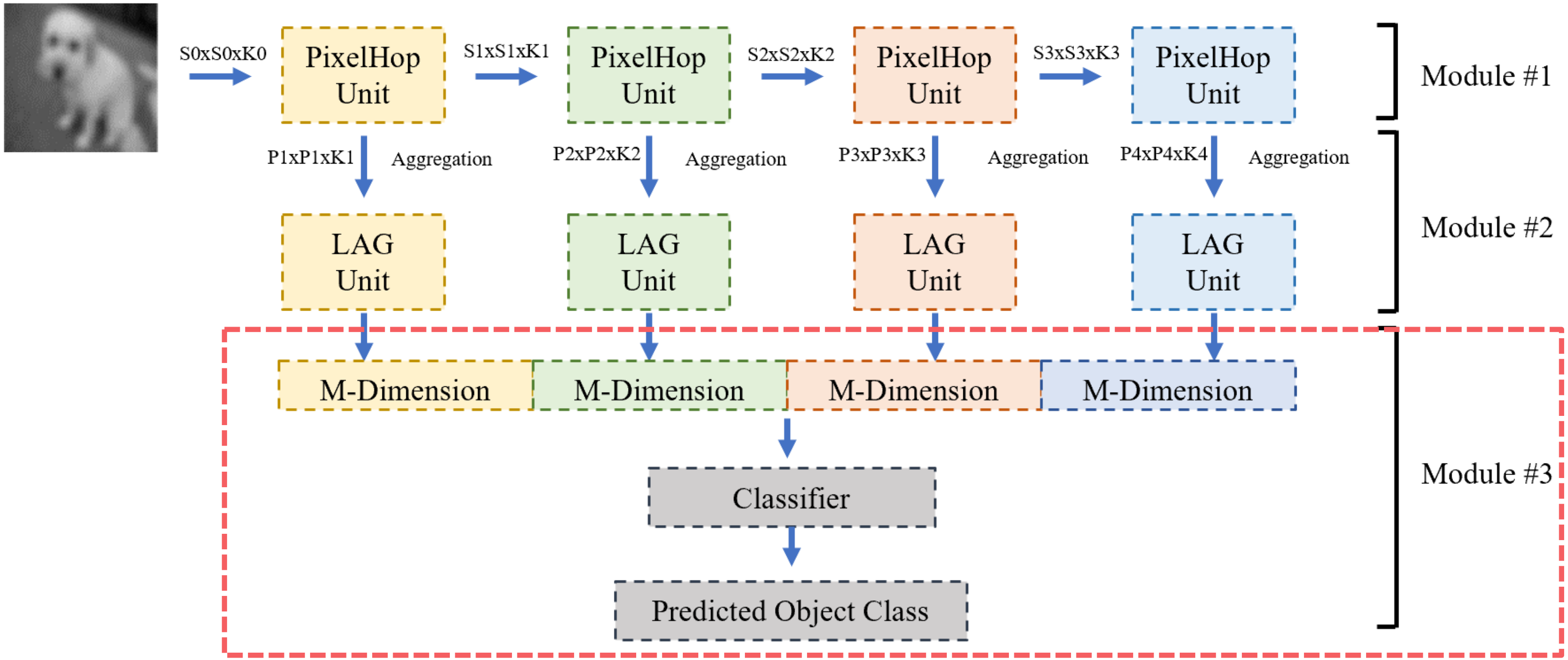
$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} & w_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & w_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{M1} & a_{M2} & \cdots & a_{Mn} & w_M \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{p}_1(\mathbf{x}) \\ \vdots \\ \mathbf{p}_j(\mathbf{x}) \\ \vdots \\ \mathbf{p}_J(\mathbf{x}) \end{bmatrix},$$



➢ Linear transform to low dimensional feature space

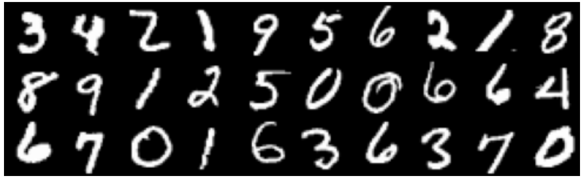
PixelHop Method

❖ Module #3: Feature concatenation across cascaded units and classification



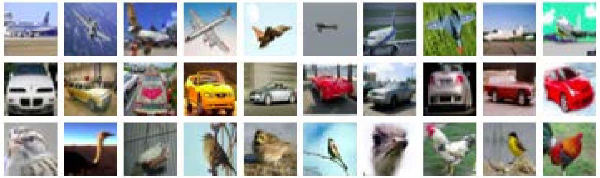
Experiment Set - up

- Datasets:
  - MNIST
    - Handwritten digits 0-9
    - Gray-scale images with size 32x32
    - Training set: 60k, Testing set: 10k
  - Fashion-MNIST
    - Gray-scale fashion images with size 32×32
    - Training set: 60k, Testing set: 10k
  - CIFAR-10
    - 10 classes of tiny RGB images with size 32×32
    - Training set: 50k, Testing set: 10k
- Evaluation:
  - Top-1 classification accuracy



MNIST

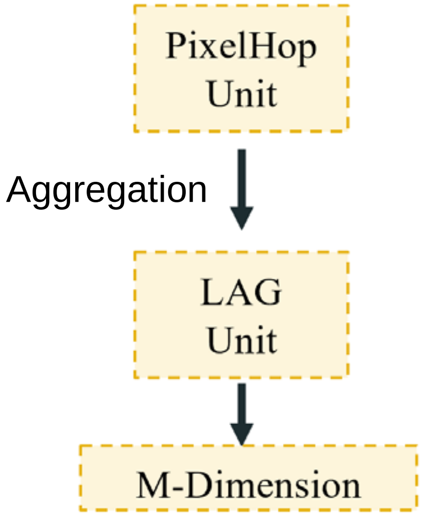
Fashion-MNIST



CIFAR-10

Experiment Set - up

- PixelHop system:
  - Four PixelHop units
  - Aggregation: mean-pooling
  - Output dimension: M = 50
  - Compute probability: α = 10
  - The multi-class SVM classifier with the Radial Basis Function (RBF) as the kernel



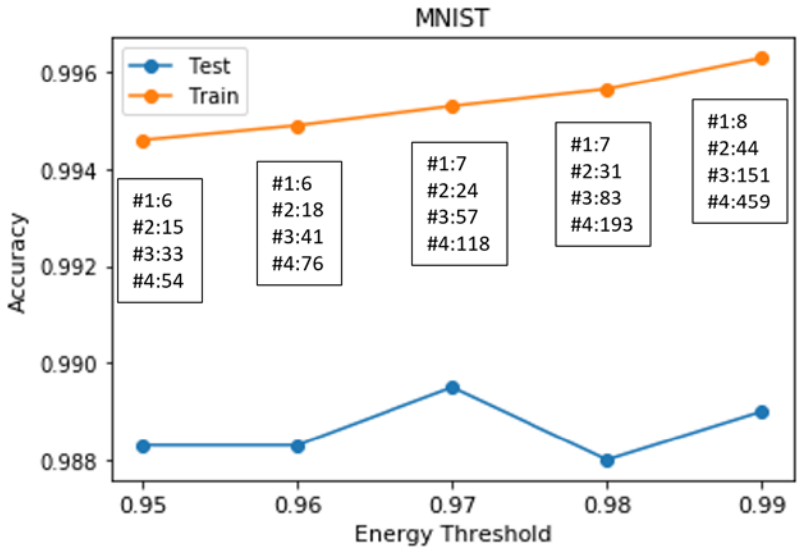
Experiment Set - up

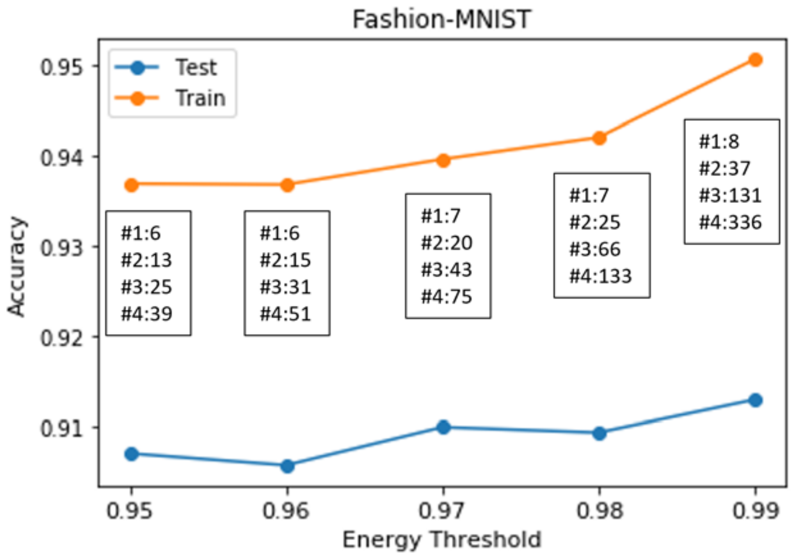
- CNN-based method:
    - Gray-scale images and color images

Architecture	Original LeNet-5	Modified LeNet-5
1st Conv Layer Kernel Size	$5 \times 5 \times 1$	$5 \times 5 \times 3$
1st Conv Layer Kernel No.	6	32
2nd Conv Layer Kernel Size	$5 \times 5 \times 6$	$5 \times 5 \times 32$
2nd Conv Layer Kernel No.	16	64
1st FC Layer Filter No.	120	200
2nd FC Layer Filter No.	84	100
Output Node No.	10	10

Results: Kernel numbers of Saab transforms

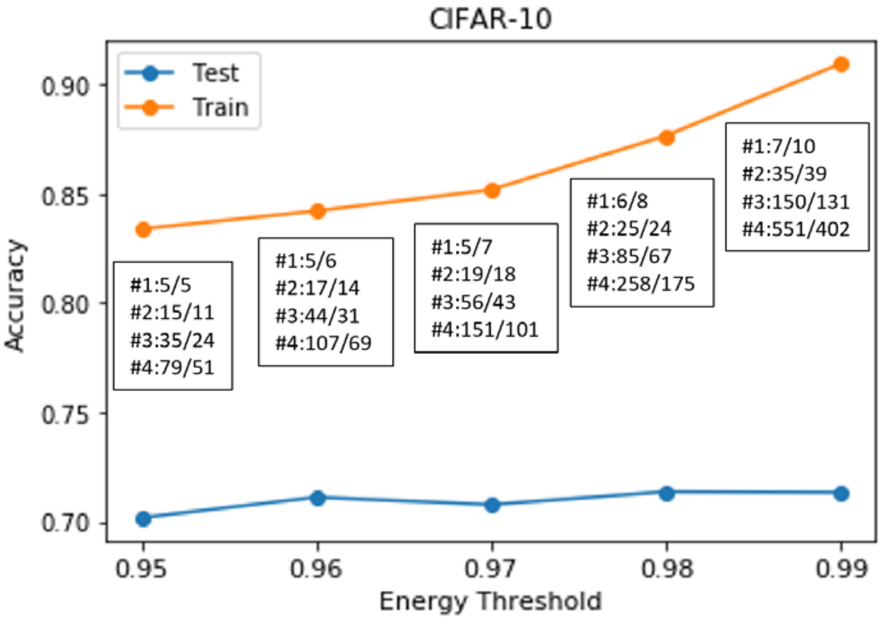
➢ Cutoff energy thresholds: 95%, 96%, 97%, 98% and 99% of the total energy





Results: Kernel numbers of Saab transforms

- Cutoff energy thresholds: 95%, 96%, 97%, 98% and 99% of the total energy



Results: Ablation Study

- Experiments on Fashion-MNIST dataset
  - Best settings: concatenation of all units, mean-pooling, SVM classifier

Feature Used		DR		Aggregation				Classifier		Test ACC (%)
ALL	Last Unit	LAG	PCA	Mean	Min	Max	Skip	SVM	RF	
	✓	✓		✓				✓		89.88
	✓		✓	✓				✓		89.11
✓		✓		✓					✓	89.31
✓		✓		✓				✓		91.30
✓		✓			✓			✓		91.16
✓		✓				✓		✓		90.83
✓		✓					✓	✓		91.14

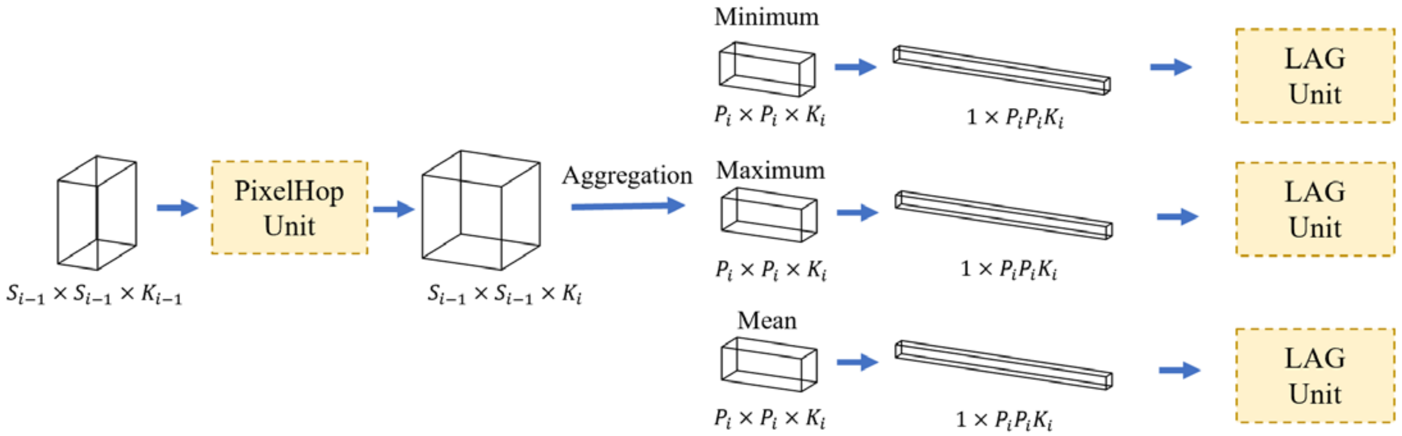
Results: Ablation Study

- Feature concatenations:
  - HOP-i: An individual representation from PixelHop unit i, i= 1,2,3,4
  - Default aggregation : concatenation of four representations
  - Advanced aggregation : concatenation of twelve representations

Dataset	HOP-1	HOP-2	HOP-3	HOP-4	default	advanced
MINST	97.00	98.35	98.45	98.71	98.90	99.09
Fashion MINST	87.38	89.35	89.96	89.88	91.30	91.68
CIFAR-10	52.27	67.86	69.08	67.91	71.37	72.66

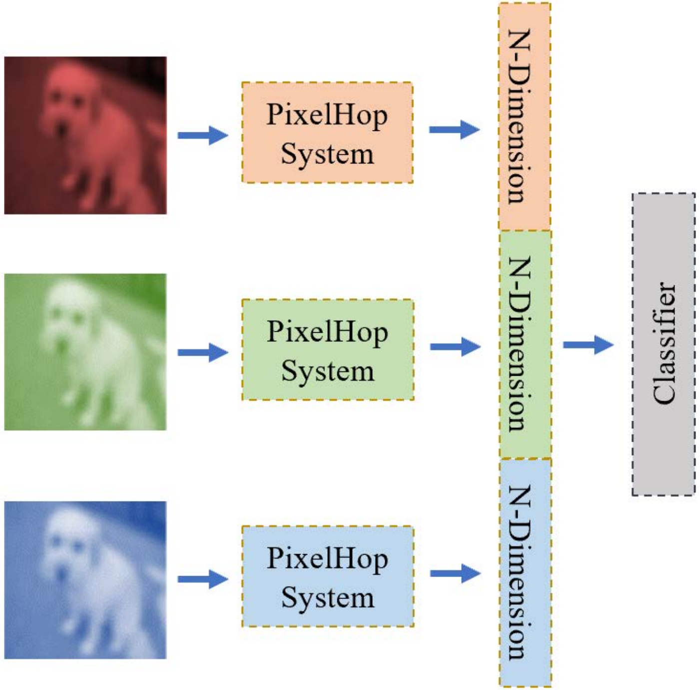
PixelHop Method

- ❖ Module #2: Aggregation and Supervised Subspace Learning
  - ➢ Aggregation: minimum, maximum, mean values



Results: CIFAR-10 (color images)

- ❖ Three color spaces: RGB, YCbCr, and Lab
- ❖ System input:
  - ➢ Three channels together
  - ➢ Single channel
  - ➢ Luminance/chrominance channel



Handle R, G, B channels separately

Results: CIFAR-10 (color images)

- System input :
  - Three channels together: RGB,YCbCr, and Lab
  - Single channel : R, G, B
  - Luminance/chrominance channel: Y,CbCr and L, ab

	RGB	R,G,B	YCbCr	Y,CbCr	Lab	L,ab
Test	68.90	69.96	68.74	71.05	67.05	**71.37**
Train	84.11	85.06	84.05	86.03	87.46	**87.65**

Results: Accuracy and Efficiency

- Comparison with LeNet-5 method

Method	MNIST	Fashion MNIST	CIFAR-10
LeNet-5	99.04	91.08	68.72
PixelHop	98.90	91.30	71.37
PixelHop$^+$	**99.09**	**91.68**	**72.66**

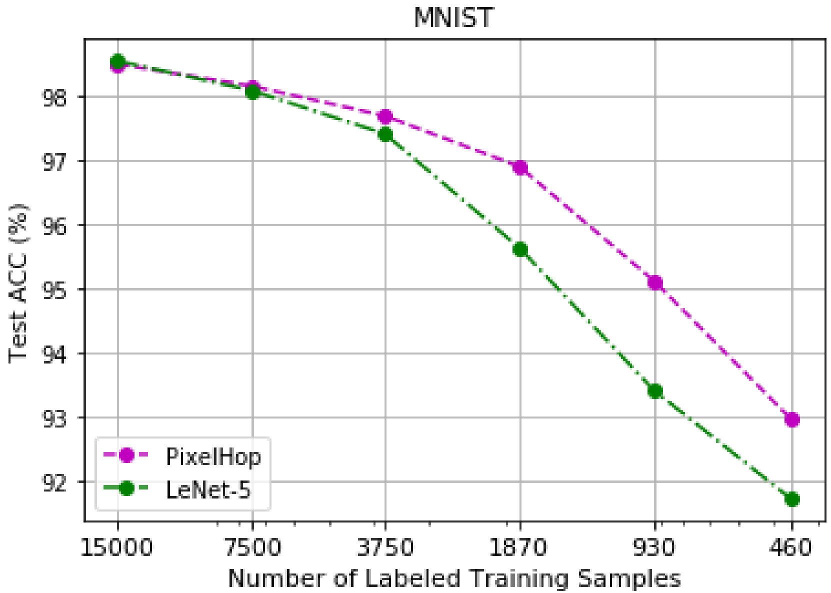
Testing accuracy (%)

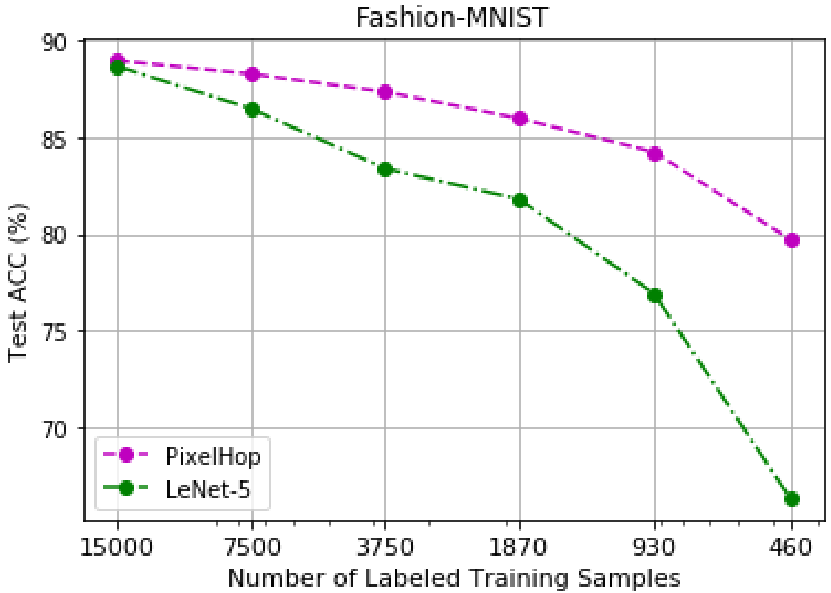
Method	MNIST	Fashion MNIST	CIFAR-10
LeNet-5	~25 min	~25 min	~45 min
PixelHop	~15 min	~15 min	~30 min

Training time

Results: Scalability

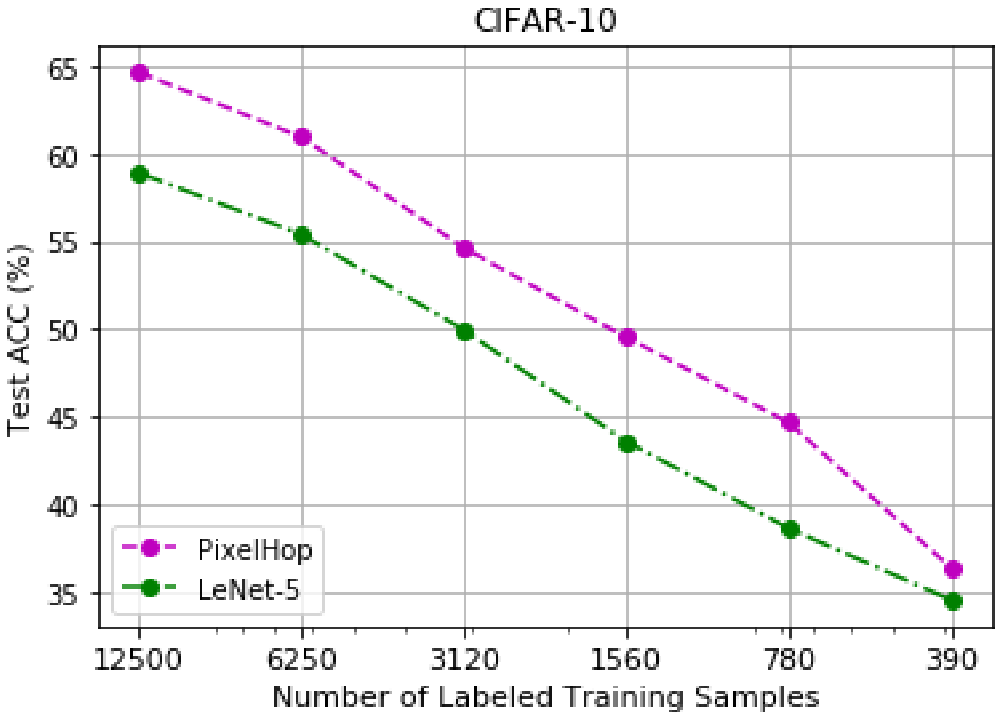
❖ Labeled training set: 1/4, 1/8, …, 1/128 of whole training set





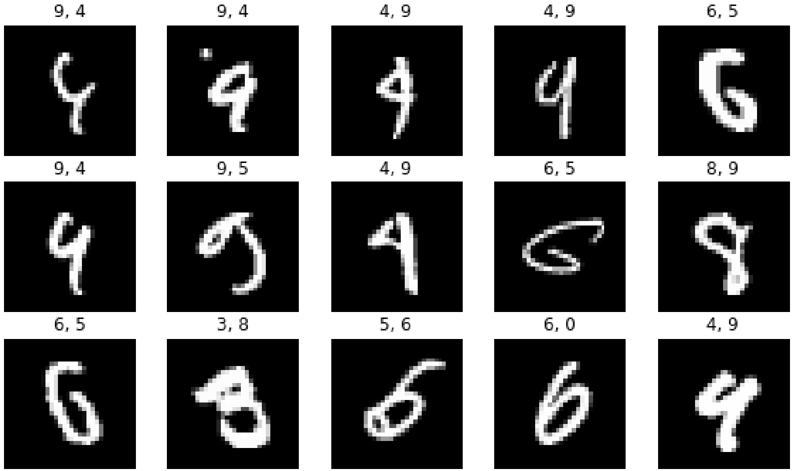
Results: Scalability

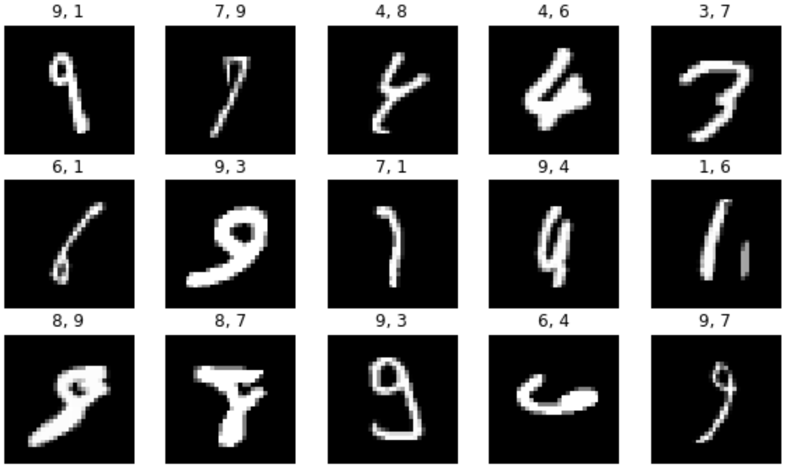
❖ Labeled training set: 1/4, 1/8, …, 1/128 of whole training set



Error Analysis: MNIST

- Challenging errors
- The rule-based method



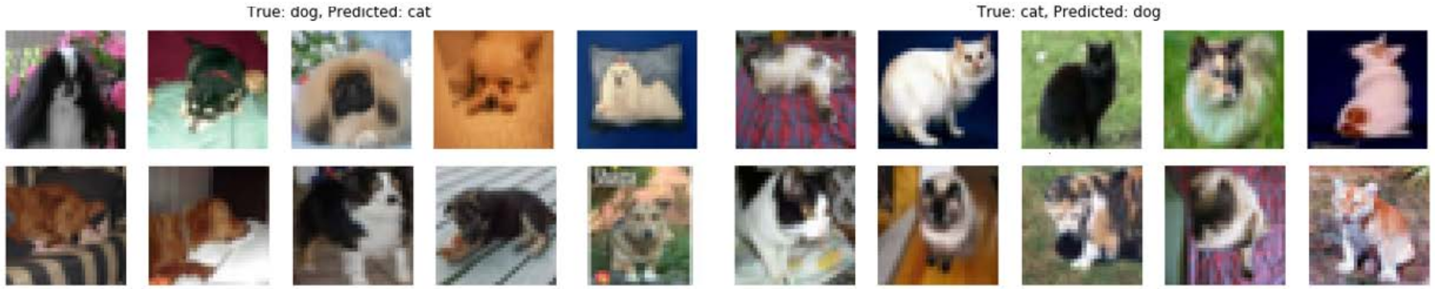


Error Analysis: Fashion - MNIST

- Hardest class: “Shirt”

Error Analysis: CIFAR - 10

- Cat vs. Dog



- Ship vs. Airplane

Summary

- Propose an image classification method and test on three benchmarking datasets
- Develop a supervised dimension reduction technique: LAG
- Design a learning system in a one-pass feedforward manner
- Mathematically explainable and scalable learning methodology

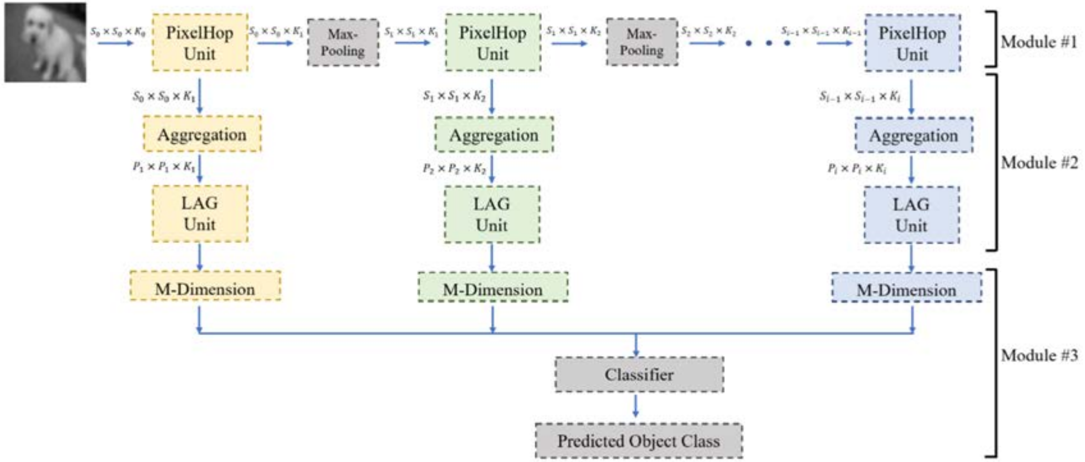
Chen, Y., & Kuo, C. C. J. (2019). PixelHop: A Successive Subspace Learning (SSL) Method for Object Recognition. Plan to submit to Journal of Visual Communication and Image Representation.

PixelHop++ Method

Background Review

Successive Subspace Learning

1. successive near-to-far neighborhood expansion in multiple stages



2. unsupervised dimension reduction via subspace approximation at each stage

3. supervised dimension reduction via label-assisted regression (LAG)

4. feature concatenation and decision making

PixelHop method for image classification

Framework of proposed PixelHop++

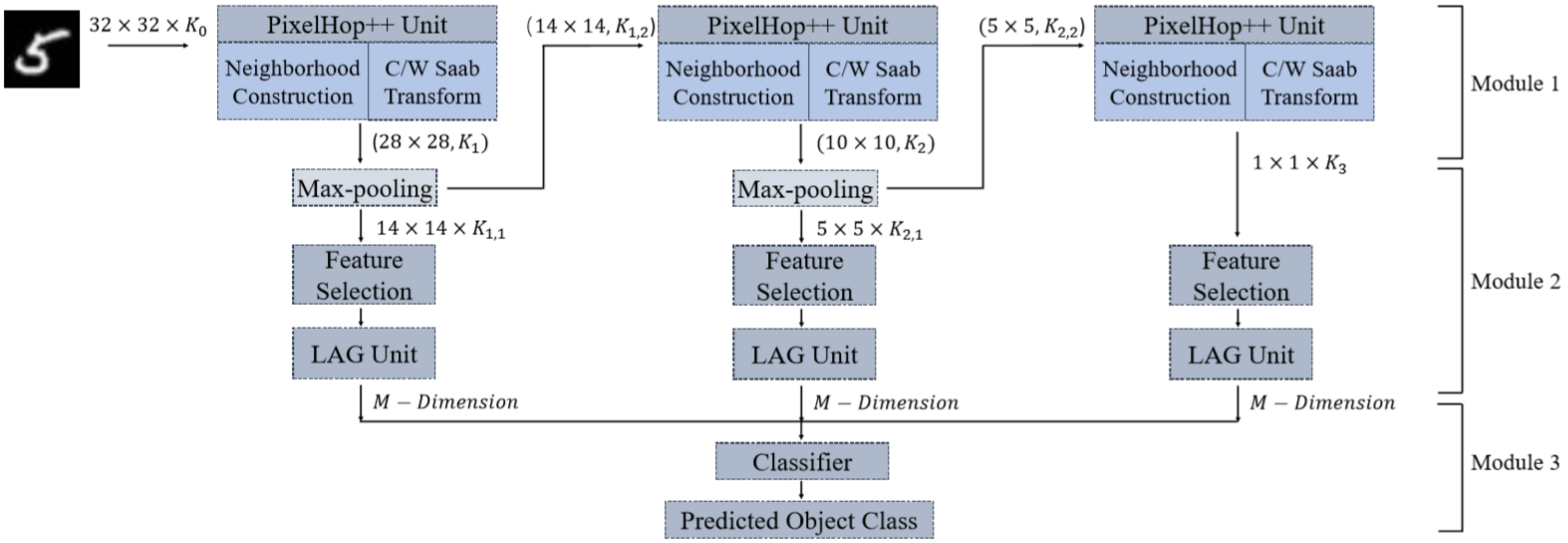


Fig. 1. The block diagram of the PixelHop++ method that contains three PixelHop++ Units in cascade.

PixelHop++ unit vs. PixelHop unit

- Channel-wise (c/w) Saab transform
  - Smaller model size
  - Provide a channel-decomposed feature tree
- Spatial-spectral separability assumption
  - Saab coefficients to be weakly correlated in the spectral domain

Table 1. Averaged correlations of filtered AC outputs from the first to the third Pixelhop units with respect to the MNIST, Fashion MNIST and CIFAR-10 datasets.

Dataset	MNIST	Fashion MNIST	CIFAR-10
Spatial 1	0.48 ± 0.05	0.51 ± 0.03	0.53 ± 0.03
Spatial 2	0.22 ± 0.03	0.29 ± 0.05	0.27 ± 0.06
Spectral 1	0.33 ± 0.07	0.12 ± 0.02	0.0156 ± 0.0005
Spectral 2	0.18 ± 0.02	0.13 ± 0.01	0.0188 ± 0.0004
Spectral 3	0.0099 ± 0.0001	0.0082 ± 0.0001	0.0079 ± 0.0004

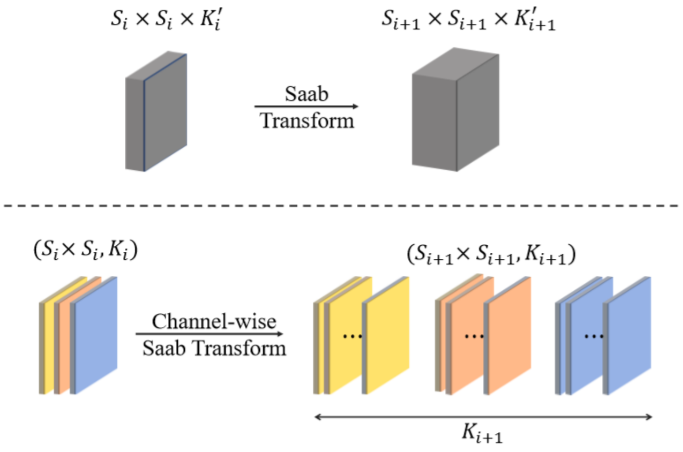


Fig. 2. Comparison of the traditional Saab transform and the proposed c/w Saab transform.

PixelHop++

- Tree-decomposed feature representation
  - Different tree levels corresponding to different receptive field sizes
  - Useful features for the image classification task
  - Unsupervised process
  - Pre-set energy threshold T to control the growth of the tree

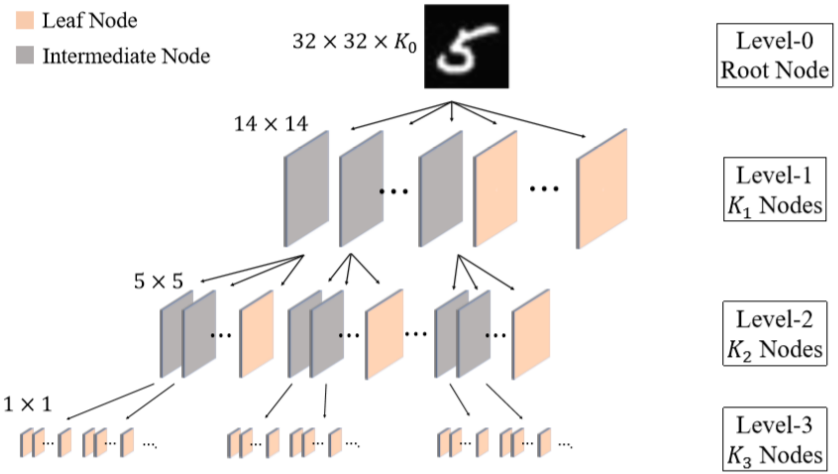


Fig. 3. Illustration of the tree-decomposed feature representation.

PixelHop++

- Module 2:
  - Cross-entropy-guided feature selection: Select top Ns features with smaller cross-entropy scores
  - Feature reduction: label-assisted regression (LAG) unit.

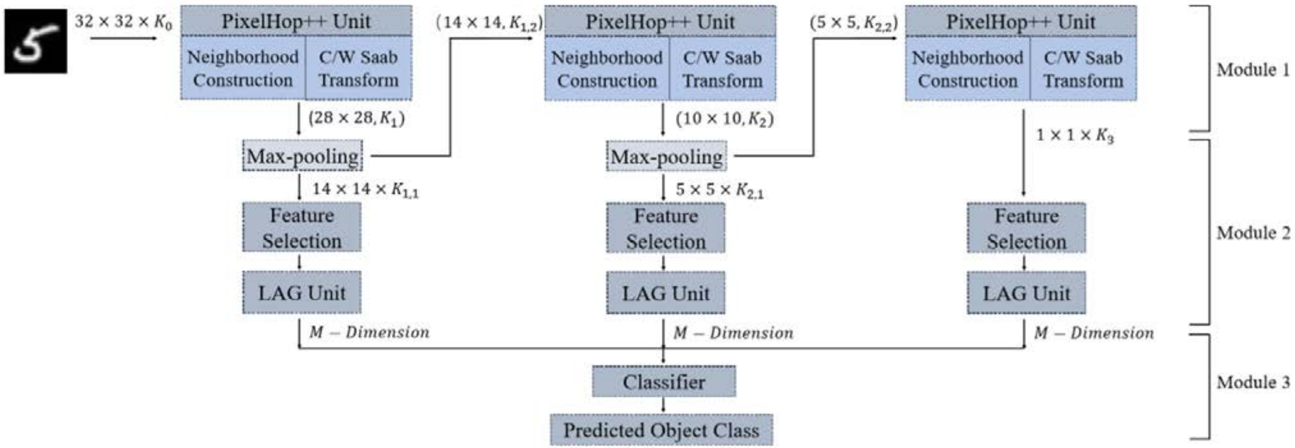


Fig. 1. The block diagram of the PixelHop++ method that contains three PixelHop++ Units in cascade.

Experimental Setups

- Datasets: MNIST, Fashion MNIST and CIFAR-10.
- Performance benchmarking of PixelHop++ and LeNet-5

Table 2. Comparison of the original and the modified LeNet-5 architectures on three benchmark dataset.

Dataset	MNIST	Fashion MNIST	CIFAR-10
1st Conv. Kernel Size	$5 \times 5 \times 1$	$5 \times 5 \times 1$	$5 \times 5 \times 3$
1st Conv. Kernel No.	6	16	32
2nd Conv. Kernel Size	$5 \times 5 \times 6$	$5 \times 5 \times 16$	$5 \times 5 \times 32$
2nd Conv. Kernel No.	16	32	64
1st FC. Filter No.	120	200	200
2nd FC. Filter No.	84	100	100
Output Node No.	10	10	10

Experimental Results

- Comparison of classification accuracy and model complexity of LeNet-5 and PixelHop++
- Two model settings of PixelHop++: a larger model and a smaller model

Table 3. Comparison of test accuracy (%) of LeNet-5 and PixelHop++ for MNIST, Fashion MNIST and CIFAR-10.

Method	MNIST	Fashion MNIST	CIFAR-10
LeNet-5	99.04	89.74	68.72
PixelHop++ (Large)	98.49	90.17	66.81
PixelHop++ (Small)	97.98	88.84	64.75

Table 4. Comparison of the model size (in terms of the total parameter numbers) of LeNet-5 and PixelHop++ for the MNIST, the Fashion MNIST and the CIFAR-10 datasets.

Method	MNIST	Fashion MNIST	CIFAR-10
LeNet-5	61,706	194,558	395,006
PixelHop++ (Large)	111,981	127,186	115,623
PixelHop++ (Small)	29,514	33,017	62,150

Experimental Results

- Effects of Hyper-Parameters in PixelHop++ model
  - Energy threshold T in Module 1
  - If the energy of a node is larger than threshold T, its response map will be further processed

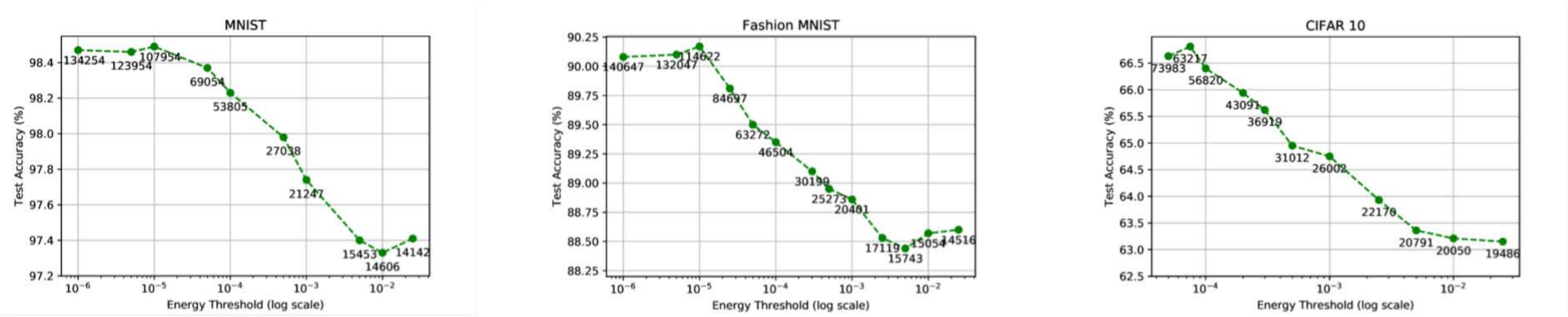


Fig. 4. The relation between the test accuracy (%) and energy threshold T in PixelHop++ for MNIST, Fashion MNIST and CIFAR-10, where the number of model parameters in Module 1 is shown at each operational point.

Experimental Results

- Effects of Hyper-Parameters in PixelHop++ model
  - Ns values in Module 2
  - Features are ordered from the smallest to the largest cross-entropy scores and the top Ns features are selected

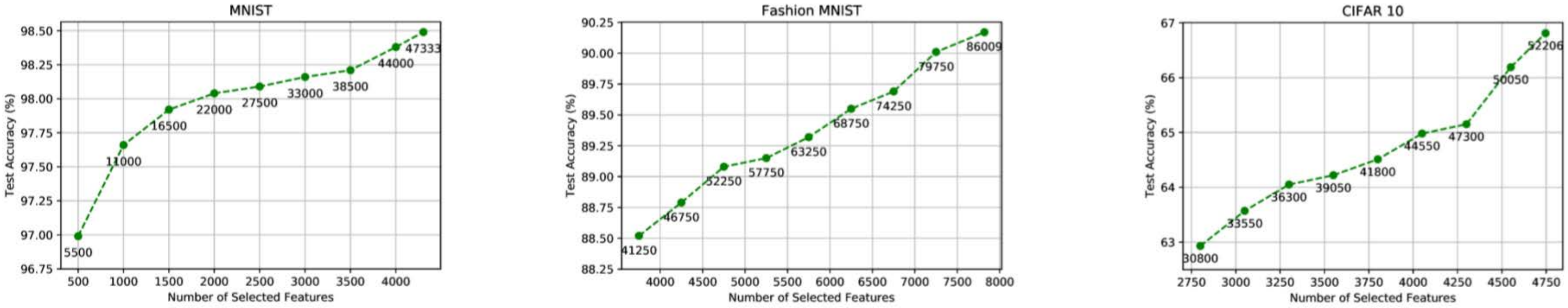
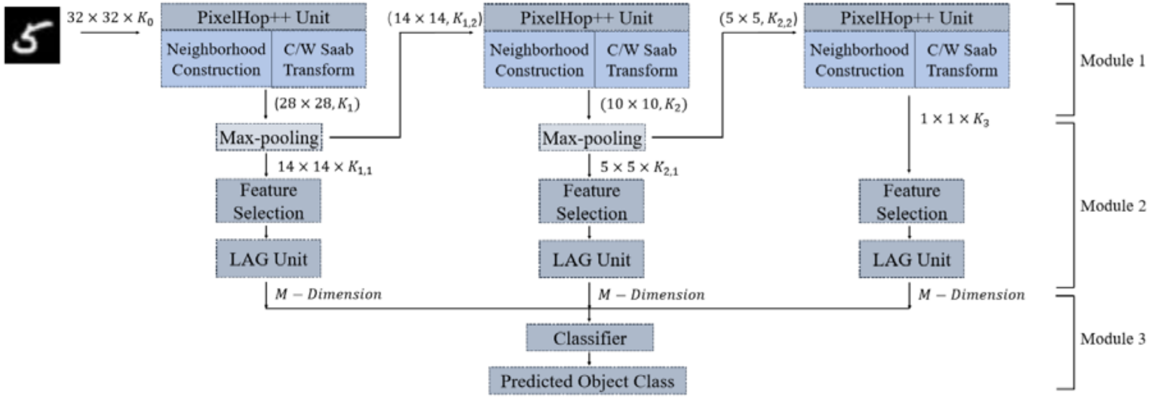


Fig. 5. The relation between test accuracy (%) and selected number N_S of cross-entropy-guided features in PixelHop++ for MNIST, Fashion MNIST and CIFAR-10, where the number of model parameters in Module 2 is shown at each operational point.

Conclusion

1. Interpretable and small learning models

2. The channel-wise Saab transform



PixelHop++ method for image classification

3. A novel tree-decomposed feature representation

4. Feature selection based on cross-entropy values

References

[1] chrome-extension://oemmndcbldboiebfnladdacbdfmadadm/http://web.media.mit.edu/~javierhr/files/slidesPCA.pdf

[2] https://github.com/zalandoresearch/fashion -mnist

[3] Yann LeCun et al., “Lenet-5, convolutional neural networks,” URL: http://yann. lecun. com/exdb/lenet, 2015.

[4] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A largescale hierarchical image database. In Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, pages 248–255. Ieee, 2009.

[6] Kuo, C. C. J., & Chen, Y. (2018). On data-driven Saak transform. Journal of Visual Communication and Image Representation, 50, 237-246.

[7]Chen, Y., Xu, Z., Cai, S., Lang, Y., & Kuo, C. C. J. (2018, June). A saak transform approach to efficient, scalable and robust handwritten digits recognition. In 2018 Picture Coding Symposium (PCS) (pp. 174-178). IEEE.

[8] Chen, Y., Yang, Y., Wang, W., & Kuo, C. C. J. (2019). Ensembles of Feedforward-Designed Convolutional Neural Networks. arXiv preprint arXiv: 1901.02154.

[9] Chen, Y., Yang, Y., & Kuo, C. C. J. (2019). Semi-Supervised Learning with Feedforward-Designed Convolutional Neural Networks

[10] Kuo, C. C. J., Zhang, M., Li, S., Duan, J., & Chen, Y. (2018). Interpretable convolutional neural networks via feedforward design. arXiv preprint arXiv:1810.02786.

[11] T. G. Dietterich. Ensemble methods in machine learning. In International workshop on multiple classifier systems, pages 1–15. Springer, 2000.

[12] https://en.wikipedia.org/wiki/Semi-supervised_learning