

# E-PixelHop: An Enhanced PixelHop Method for Object Classification

Yijing Yang, Vasileios Magoulaitis and C.-C. Jay Kuo  
University of Southern California, USA  
E-mail: {yijingya, magoulia, jckuo}@usc.edu

**Abstract**—Based on PixelHop and PixelHop++, which are recently developed using the successive subspace learning (SSL) framework, we propose an enhanced solution for object classification, called E-PixelHop, in this work. E-PixelHop consists of the following steps. First, to decouple the color channels for a color image, we apply principle component analysis and project RGB three color channels onto two principle subspaces which are processed separately for classification. Second, to address the importance of multi-scale features, we conduct pixel-level classification at each hop with various receptive fields. Third, to further improve pixel-level classification accuracy, we develop a soft-label smoothing (SLS) scheme to ensure prediction consistency. Forth, pixel-level decisions from each hop and from each color subspace are fused together for image-level decision. Fifth, to resolve confusing classes for further performance boosting, we formulate E-PixelHop as a two-stage pipeline. In the first stage, multi-class classification is performed to get a soft decision, where the top 2 classes with the highest probabilities are called confusing classes. Then, we conduct a binary classification in the second stage. The main contributions lie in Steps 1, 3 and 5. We use the classification of the CIFAR-10 dataset as an example to demonstrate the effectiveness of the above-mentioned key components of E-PixelHop.

## I. INTRODUCTION

Object classification has been studied for many years as a fundamental problem in computer vision. With the development of convolutional neural networks (CNNs) and the availability of larger scale datasets, we see a rapid success in the classification using deep learning for both low- and high-resolution images [1], [2], [3], [4], [5]. Deep learning networks use backpropagation to optimize an objective function to find the optimal parameters of networks. Although being effective, deep learning demands a high computational cost. As the network goes deeper, the model size increases dramatically. One major challenge associated with deep learning is that its underlying mechanism is not transparent.

Recently, based on the successive subspace learning (SSL) framework, the PixelHop [6] and the PixelHop++ [7] methods have been proposed for image classification. Both follow the traditional pattern recognition paradigm and partition the classification problem into two cascaded modules: 1) feature extraction and 2) classification. Powerful spatial-spectral features can be extracted from PixelHop and PixelHop++ in an unsupervised manner. Then, they are fed into a trained classifier for final decision. Every step in PixelHop/PixelHop++ is explainable, and the whole solution is mathematically transparent.

In this work, we propose an enhanced PixelHop method, named E-PixelHop. It consists of the following main steps.

- 1) To decouple the color channels for a color image, E-PixelHop applies principle component analysis (PCA) and project RGB three color channels onto two principle subspaces which are processed separately for classification.
- 2) To address the importance of multi-scale features, we conduct pixel-level classification at each hop which corresponds to a patch of various sizes in the input image.
- 3) To further improve pixel-level classification accuracy, we develop a soft-label smoothing (SLS) scheme to ensure intra-hop and inter-hop pixel-level prediction consistency.
- 4) Pixel-level decisions from each hop and from each color subspace are fused together for image-level decision.
- 5) To resolve confusing classes for further performance boosting, we formulate E-PixelHop as a two-stage pipeline. In the first stage, multi-class classification is performed to get a soft decision for each class, where the top 2 classes with the highest probabilities are called confusing classes.

The main contributions of E-PixelHop lie in Steps 1, 3 and 5.

The rest of this paper is organized as follows. Related work is reviewed in Sec. II. The E-PixelHop method is presented in Sec. III. The label smoothing procedure is detailed in Sec. IV. Experimental setup and results are detailed in Sec. V. Finally, concluding remarks and future research directions are drawn in Sec. VI.

## II. REVIEW OF RELATED WORK

### A. Multi-scale Features for Object Classification

Handcrafted features were extracted before the deep learning era. To obtain multi-scale features, Mutch *et al.* [8] applied Gabor filters to all positions and scales. Scale-invariant features can be derived by alternating template matching and max-pooling operations. Schnitzspan *et al.* [9] proposed a hierarchical random field that combines the global-feature-based methods with the local-feature-based approaches in one consistent multi-layer framework. As the basis of convolutional neural networks, Neocognitron [10] builds a hierarchical model for shape processing following the visual system in order to get translation-invariant features. Being inspired by

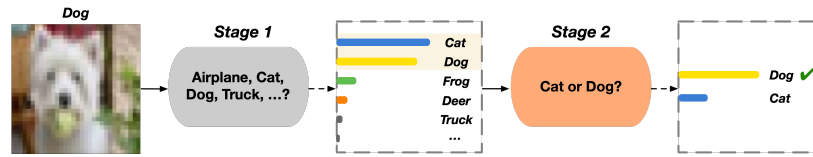


Fig. 1. E-PixelHop is a two-stage sequential classification method.

biology, Serre *et al.* [11] proposed a hierarchical system that follows the structure of visual cortex for feature representation.

The classification decision in deep learning is usually made using features from the deepest convolutional layer. Recently, more investigations are made to exploit outputs from shallower layers to improve the classification performance. For example, Liu *et al.* [12] proposed a cross-convolutional-layer pooling operation that extracts local features from one convolutional layer and pools the extracted features with the guidance of the next convolutional layer. Jetley *et al.* [13] extracted features from shallower layers and combined them with global features. The joint information is used to estimate attention maps for further classification performance improvement.

### B. Hierarchical Classification Strategy

It is easier to distinguish between classes of dissimilarity than those of similarity. For example, one should distinguish between cats and cars better than between cats and dogs. Along this line, one can build a hierarchical relation among multiple classes based on their semantic meaning to improve classification performance [14], [15], [16]. Alsallakh *et al.* [17] investigated ways to exploit the hierarchical structure to improve classification accuracy of CNNs. Yan *et al.* [18] proposed a hierarchical deep CNN (HD-CNNs) which embeds deep CNNs into a category hierarchy. It separates easy classes using a coarse category classifier and distinguishes difficult classes with a fine category classifier. Chen *et al.* [19] proposed to merge images associated with confusing anchor vectors into a confusion set and split the set to create multiple subsets in an unsupervised manner. A random forest classifier is then trained for each confusion subset to boost the scene classification performance. An identification and resolution scheme of confusing categories was proposed by Li *et al.* [20] based on binary-tree-structured clustering. Zhu *et al.* [21] defined different levels of categories and proposed a Branch Convolutional Neural Network (B-CNN) that outputs multiple predictions and ordered them from coarse to fine along concatenated convolutional layers.

### C. Successive Subspace Learning

Being inspired by deep learning, the successive subspace learning (SSL) methodology was proposed by Kuo *et al.* in a sequence of papers [22], [23], [24], [25]. SSL-based methods learn feature representations in an unsupervised feedforward manner using multi-stage principle component analysis (PCA). Joint spatial-spectral representations are obtained at different scales through multi-stage transforms. Three variants of the PCA transform were developed. They are the Saak (subspace

approximation with augmented kernels) transform [24], [26], the Saab (subspace approximation via adjusted bias) transform [25], and the channel-wise (c/w) Saab transform [7]. The c/w Saab transform is the most effective one among the three since it can reduce the model size and improve the transform efficiency by learning filters in each channel separately. This is achieved by exploiting the weak correlation between different spectral components in the Saab transform. These features can be used to train classifiers in the training phase and provide inference in the test phase. Two image classification pipelines, PixelHop [6] using the Saab transform and PixelHop++ [7] using the c/w Saab transform, were designed. Since the feature extraction module is unsupervised, a label-assisted regression (LAG) unit was proposed in PixelHop as a feature transformation, aiming at projecting the unsupervised feature to a more separable subspace. In PixelHop++, cross-entropy based feature selection is applied before each LAG unit to select task-dependent features, which provides a flexible tradeoff between model size and accuracy. Ensemble methods were used in [27] to boost the classification performance. SSL has been successfully applied to many other classification tasks, for example, point cloud classification [28], [29], face gender classification [30], biomedical image classification [31], etc.

## III. E-PIXELHOP METHOD

The E-PixelHop method for object classification is proposed in this section. An overview of E-PixelHop is described in Sec. III-A. Then, a multi-class classification baseline is described in Sec. III-B. Finally, confusion class resolution is presented in Sec. III-C.

### A. System Overview

The overall framework of the E-PixelHop is illustrated in Fig. 1. It is a two-stage sequential classification method. The first stage serves as a baseline that performs classification among all object classes. Its output is a set of soft decision scores that indicate the probability of each class. The classes with the highest  $M$  probabilities for each image as its confusing group. Typically, we set  $M = 2$ . In the second stage, a one-versus-one competition is conducted to refine the prediction result. The two-stage sequential decision makes a coarse prediction and, then, focuses on confusing classes resolution for more accurate prediction.

We follow the traditional pattern recognition paradigm by dividing the problem into feature extraction and classification two separate modules. Multi-hop c/w Saab transforms are used to extract joint spatial-spectral features in an unsupervised manner. For classification, we adopt pixel-based classification

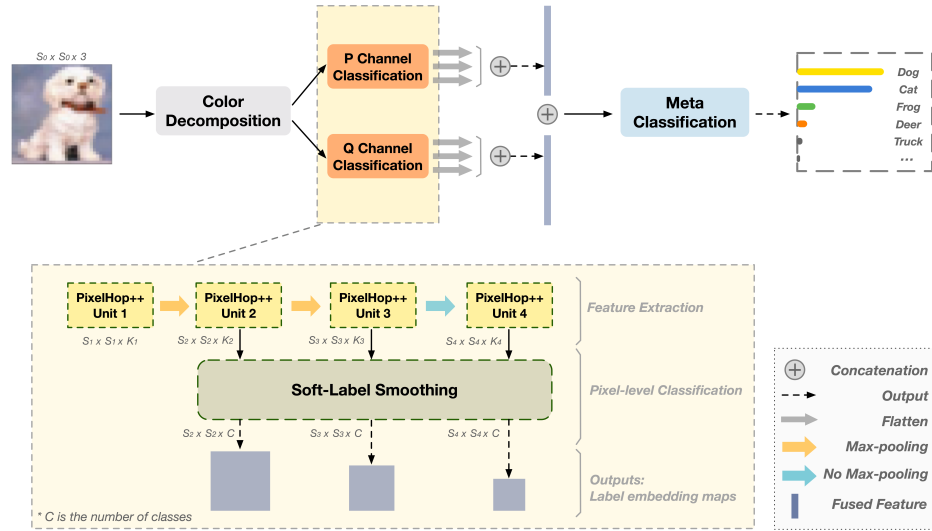


Fig. 2. Illustration of the E-PixelHop classification baseline.

in each hop that predicts objects of different scales at different spatial locations, where a pixel in a deeper hop denotes a patch of a larger size in the input image. In the training, ground truth labels at pixels follow image labels. Pixel-based classification based on the intra-hop information only tends to be noisy. To address this problem, a soft-label smoothing (SLS) method is proposed to reduce prediction uncertainty.

### B. Stage 1: Multi-class Baseline Classifier

We use the CIFAR-10 dataset [32] as an example to explain the architecture of the multi-class classification baseline. It consists of four modules: 1) Color decomposition, 2) PixelHop++ feature extraction, 3) Pixel-level classification with label smoothing, 4) Meta classification. The system diagram for the baseline classification is shown in Fig. 2.

**1) Color decomposition.** For input color images of RGB three channels, the feature dimension is three times of the spatial dimension in a local neighborhood. To reduce the dimension, we apply the principle component analysis (PCA) to the 3D color channels of all pixels. That is, we collect RGB 3D color vectors from pixels in all input images and learn the PCA kernels from collected samples. They project the RGB color coordinates to decoupled color coordinates, which are named the *PQR* color coordinates, where P and Q channels correspond to the 1<sup>st</sup> and 2<sup>nd</sup> principle components. Experimental results show that P and Q channels contain approximately 98.5% of the energy of RGB three channels. Specifically, the P channel has 91.08% and the Q channel 7.38%. Since P and Q channels are uncorrelated with each other, we can proceed them separately in modules 2 and 3.

**2) PixelHop++ feature extraction.** As shown in Fig. 2, we use multiple PixelHop++ units [7] to extract features from P and Q channels, respectively. One PixelHop++ unit consists of two cascaded operations: 1) neighborhood construction with a defined window size in the spatial domain, and 2) the channel-wise (c/w) Saab transform. The input to the first PixelHop++

unit is a tensor of dimension  $(S_0 \times S_0) \times K_0$ , where  $S_0 \times S_0$  is the spatial dimension,  $K_0$  is the spectral dimension, and subscript 0 indicates that it is the hop-0 representation. For raw RGB images,  $K_0 = 3$ . Through color transformation, we have  $K_0 = 1$  for each individual P or Q channel. The output of the  $i$ -th PixelHop++ unit is denoted by  $(S_i \times S_i) \times K_i$ , which is the hop- $i$  representation. Instead of learning the filters of the convolutional layers in CNNs through end-to-end optimization with backpropagation, c/w Saab transform learns the  $K_i$  filters by studying the spatial correlations in the constructed neighborhood in each channel at hop- $i$ , as a variant of PCA. Different c/w Saab filters represent different frequency bands. The filter with a larger eigenvalue corresponds to a lower frequency component. The hyper parameter  $K_i$  is determined by a pre-set energy threshold representing the percentage of the total energy to be kept in each PCA. The more energy is kept, the higher resulted  $K_i$  filters are expected. As  $i$  increases, the receptive field associated with a pixel in hop- $i$  becomes larger and more global information of the input image is captured. To enable faster expansion of the receptive field, we apply max-pooling with stride 2 in shallower hops.

**3) Pixel-level classification with label smoothing.** The output from each PixelHop++ unit can be viewed as a feature tensor. The feature vector associated with a pixel location is fed into a pixel-level classifier to get a probability vector whose element indicates the probability of belonging to a certain object class. This probability vector can be viewed as a soft decision or a soft label. The soft label can be noisy, and we propose a soft-label smoothing mechanism to make the soft decision at neighboring spatial locations and hops more consistent. This is one of the main contributions of E-PixelHop. It will be elaborated in Sec. IV.

**4) Meta classification.** The soft decision maps from the pixel-level classification in different hops of P and Q channels are flattened and concatenated as a single feature vector for

each image. It is used to train a meta classifier for the final image-level decision which gives the baseline prediction.

### C. Stage 2: Confusion Set Resolution

Instead of constructing a hierarchical learning structure manually as done in [21], E-PixelHop identifies confusion sets using the predictions of the baseline classifier in Stage 1. The baseline classifier outputs a  $C$ -dimensional soft decision vector for each image. Generally, the  $M$  classes with the highest probabilities can define a confusion set. For a  $C$ -class classification problem, we have at most  $N_{cg} = \frac{C!}{M!(C-M)!}$  confusion sets. In practice, only a portion of  $N_{cg}$  sets contribute to the performance gain significantly since the member of some confusion set is few. In the following, we focus on the case with  $M = 2$ .

Without loss of generality, we use the ‘‘Cat and Dog’’, the two most confusing classes from CIFAR-10, as an example to explain our approach for confusion resolution. In the training, we collect all 5,000 Cat images and 5,000 Dog images from the training set. For test images that have the top-2 candidates of Cat or Dog, we include them in the Cat/Dog confusion set. If an image belongs to this set yet its ground truth class is neither Cat nor Dog, its misclassification cannot be corrected. As a result, the top-2 accuracy of the baseline classifier offers an upper bound for the final performance. For binary classification, the PixelHop++ features learned from the 10 classes in the baseline for P and Q channels are reused. Since the label is a 2-D vector whose element sum is equal to unity, we can simplify the label to a scalar in label smoothing. Thus, the lab map has a size of  $S_i \times S_i \times 1$  at hop- $i$ . We use the same ensemble scheme to fuse P and Q predicted labels. The meta classifier finally yields a binary decision for each test image in the confusion set to be a dog or a cat.

## IV. SOFT-LABEL SMOOTHING

Soft-label prediction at each pixel of a certain hop tends to be noisy. Label smoothing attempts to convert noisy decision to a clean one using adjacent label prediction from the same hop or from adjacent hops. The uncertainty of intra-hop prediction is first discussed in Sec. IV-A. The soft-label smoothing (SLS) scheme is then presented in Sec. IV-B.

### A. Intra-Hop Prediction

For image classification, only the global coarse-scale information may not be sufficient. Mid-range to local fine-scale information can help characterize discriminant information of a region of interest. However, the object size varies a lot from images to images. Different receptive fields are needed for various object sizes and the context around objects. To exploit the characteristics of different scales, we do pixel-wise classification at each hop and ensemble predicted class probabilities from all hops. One naive approach is to use the extracted c/w Saab features of a single hop for classification, called intra-hop prediction. For example, c/w Saab features at hop- $i$  are of dimension  $(S_i \times S_i) \times K_i$ , where  $(S_i \times S_i)$  and  $K_i$  represent spatial and spectral dimensions, respectively.

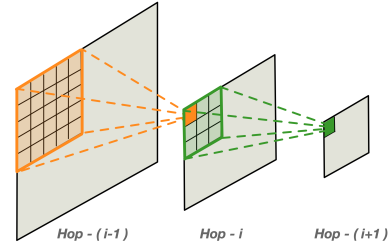


Fig. 3. Illustration of local graph construction based on c/w Saab transforms.

$N$  training images contribute  $N \times S_i \times S_i$  training samples for intra-hop prediction. Each sample has a feature vector of dimension  $K_i$ . For the  $n$ -th training image, all  $S_i \times S_i$  pixels share the same object label  $y_n$ . For the test image, the output of intra-hop prediction at hop- $i$  is a tensor of predicted class probabilities with the same spatial resolution of the input. The class probability vectors are soft decision labels of a pixel at hop- $i$ , which corresponds to a spatial region of the input image with its size equal to the receptive field of the pixel.

Pixel-wise classification can be noisy since its training label is set to the image label. For a dog image, only part of the image contains discriminant dog characteristics while other part belongs to background and has nothing to do with dogs. On one hand, the shared background of dog and cat images will have two different labels in the training so that its prediction of dog or cat becomes less confident. On the other hand, the intra-hop classifier will be more confident in its predictions in regions that are associated with a particular class uniquely. For example, for a binary classification between a cat and a dog, contents in background such as grass and sofa are usually common so that the classifier is less confident in these regions as compared with the foreground regions that contain cat/dog faces.

### B. Soft-Label Smoothing (SLS)

Since the decision at each pixel is made independently of others in pixel-level classification, there are local prediction fluctuations. We expect consistency in predictions in a spatial neighborhood as well as regions of a similar receptive field across hops that form a pyramid. To reduce the prediction fluctuation, we propose a soft-label smoothing (SLS) scheme built upon pixel-level classification which iteratively uses predicted labels to enhance classification performance while keeping intra-hop and inter-hop consistency. It consists of three components: a) local graph construction; b) label initialization; and c) iterative cross-hop label update. A similar high-level idea was proposed in GraphHop [33], yet the details are different.

a) *Local Graph Construction*: The tree-decomposed-structure of multi-Hop c/w Saab transforms corresponds to a directed graph. Each spatial location is a node. As shown in Fig. 3, we define three node types – child nodes, parent nodes, and sibling nodes. Hop- $i$  features are generated by an input of size  $5 \times 5$  through the c/w Saab transform from hop- $(i-1)$ . The  $5 \times 5$  locations in hop- $(i-1)$  form 25 child nodes for the orange node at hop- $i$ . The orange node is then covered by

a window of size  $3 \times 3$  to generate the green node in hop- $(i+1)$ . Thus, it is one of the 9 child nodes of the green node and the green node is the parent node of the orange one. As to sibling nodes, they are neighboring pixels at the same hop. Specifically, we consider eight nearest neighbors. The union of the eight neighbors and the center pixel form a  $3 \times 3$  window.

*b) Label Initialization:* For the hop- $i$  feature tensor of dimension  $S_i \times S_i \times K_i$ , it consists of  $S_i^2$  nodes. We treat each node at the same hop as a sample and train a classifier to generate a soft decision as its initial label. In the training, the image-level object class is propagated to all nodes at all hops. Only the c/w Saab features are used for label prediction initially, which is given by

$$\mathbf{Z}_{i,u,v}^{(0)} = h_i^{(0)}(\mathbf{f}_{i,u,v}), \quad (1)$$

where  $\mathbf{Z}_{i,u,v}^{(0)}$  is the initial label prediction for node  $(u, v)$  at hop- $i$ ,  $\mathbf{f}_{i,u,v}$  is the corresponding c/w Saab feature vector, and  $h_i^{(0)}$  denotes the pixel-level classifier, whose pixel-level labels are propagated from the image-level labels. For the hops whose child nodes have label initialization, we concatenate the c/w Saab features and the label predicted in the previous hop to train the classifier. That is,

$$\mathbf{Z}_{i,u,v}^{(0)} = h_i^{(0)}(\mathbf{f}_{i,u,v} \oplus \mathbf{Z}_{i-1,u,v}^{(0)}), \quad (2)$$

where  $\oplus$  represents the concatenation operation between two sets of feature vectors. Here, the initial predicted label of  $M \times N$  child nodes constructed by the local graph for node  $(u, v)$  are averaged as

$$\mathbf{Z}_{i-1,u,v}^{(0)} = \frac{1}{M \times N} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \mathbf{Z}_{i-1,m,n}^{(0)}. \quad (3)$$

This is different from the intra-hop prediction because of the aggregation with child nodes and the inter-hop information is exchanged.

*c) Cross-hop Label Update:* Besides the intra-hop label update as described above, we conduct the inter-hop label update iteratively. The label is updated from the shallow hop to the deep hop as shown in Fig. 4 at each iteration. To update the label at the yellow location in hop- $i$ , we gather the neighborhood that forms a pyramid. The labels are averaged among sibling and child nodes in blue and green regions, respectively. They are aggregated with labels of its parent node and itself. For a  $C$ -class classification scenario, the degree of freedom of the predicted soft-label is  $C-1$  so that the aggregated label dimension is equal to  $4 \times (C-1)$ . For example, in the Cat/Dog confusion set with  $C=2$ , only the dimension corresponding to dog in the current label is extracted. This gives the aggregated feature of dimension 4-D. To control the model size, we do not use the c/w Saab feature for label update. A pixel-level classifier  $h_i^{(k-1)}$  is trained against the aggregated label dimension regarded as the input features. The output soft decisions are the updated labels at iteration  $k$  in form of

$$\mathbf{Z}_{i,u,v}^{(k)} = h_i^{(k-1)}(\mathbf{Z}_{agg,i,u,v}^{(k-1)}), \quad (4)$$

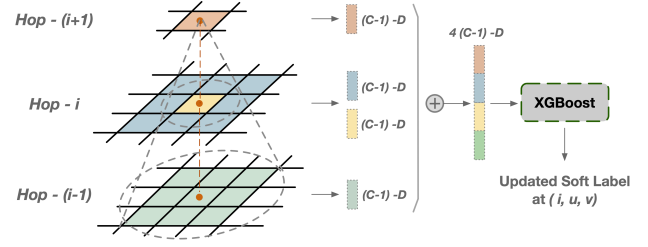


Fig. 4. Illustration of cross-hop label update.

where  $\mathbf{Z}_{agg,i,u,v}^{(k-1)}$  denotes the aggregated features of the cross-hop neighborhood.

## V. EXPERIMENTS

### A. Experimental Setup

We evaluate the classification performance of E-PixelHop using the CIFAR-10 dataset that contains 10 classes of tiny color images of spatial resolution  $32 \times 32$ . There are 50,000 training images and 10,000 test images. The hyper parameters of the PixelHop++ feature extraction architecture are listed in Table I. The spectral dimensions are decided by energy thresholds in the c/w Saab transform.

For baseline classification in stage-1 and confusion class resolution in stage-2, we adopt different pixel-based prediction schemes. For stage-1, pixel-based prediction is conducted among three output feature maps, namely, hop-2 after max-pooling, hop-3 and hop-4. The SLS method without cross-hop label update is performed among these three hops. For stage-2, besides hop-3 and hop-4, we consider hop-2 without max-pooling for higher resolution to leverage more local information in distinguishing confusion classes. The full SLS method is conducted as the pixel-based prediction in stage-2. Yet, hop-1 features are not used since their receptive field is too small. We use the extreme gradient boosting (XGBoost) classifier for all classification tasks in E-PixelHop.

By following [2], [4], we adopt data augmentation in both training and test for performance improvement. It includes original input images, random squared/rectangular croppings, contrast manipulation, horizontal flipping. Eight variants of each input are created. The Lanczos interpolation is applied to cropped images to resize them back to the  $32 \times 32$  resolution. In testing phase, the classification soft decisions are averaged among the eight variants to get the final score for a test image, thus yielding a more robust result.

TABLE I  
HYPER PARAMETERS OF THE PIXELHOP++ FEATURE EXTRACTION ARCHITECTURE.

	Filter		Output Feature	
	Window Size	Stride	Spatial	Spectral (P/Q)
Hop-1	$5 \times 5$ (pad 2)	$1 \times 1$	$32 \times 32$	24 / 22
Max-pooling 1	$3 \times 3$	$2 \times 2$	$15 \times 15$	24 / 22
Hop-2	$5 \times 5$ (pad 2)	$1 \times 1$	$15 \times 15$	144 / 114
Max-pooling 2	$3 \times 3$	$2 \times 2$	$7 \times 7$	144 / 114
Hop-3	$3 \times 3$	$1 \times 1$	$5 \times 5$	203 / 174
Hop-4	$3 \times 3$	$1 \times 1$	$3 \times 3$	211 / 185



TABLE II  
COMPARISON OF THE FEATURE DIMENSION AND TEST ACCURACY (%) USING DIFFERENT COLOR SPACES IN THE BASELINE CLASSIFICATION.

Color Space	Feature Dimension			Test Accuracy
	Hop-2	Hop-3	Hop-4	
<i>P</i>	100	142	148	70.26
<i>Q</i>	80	122	129	58.8
<i>PQ-Ensemble</i>	(180)	(264)	(277)	<b>73.72</b>
<i>RGB</i>	183	246	271	73.42

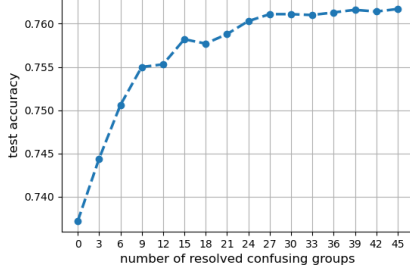


Fig. 5. The plot of CIFAR-10 test accuracy (%) as a function of the cumulative number of resolved confusion sets.

### B. Effects of Color Decomposition

E-PixelHop conducts classification in *P* and *Q* channels separately and then ensembles the classification results as shown in Fig. 2. We compare different ways to handle color channels for the baseline classifier in Table II. Since the *Q* channel contains much less energy than the *P* channel, its classification performance is worse. The ensemble of *P* and *Q* channels outperforms the *P* channel alone by 3.46% in top-1 accuracies. It shows that *P* and *Q* channels are complementary to each other. Furthermore, since the image information represented by different color spaces is expected to be similar, we control the feature dimension from RGB and PQR color spaces to be similar with each other after ANOVA-based feature selection. Specifically, the dimension for PQ-Ensemble is the summation of those in both *P* and *Q* channels. It is shown that PQ-Ensemble gives a slightly better performance than RGB but with much lower complexity. This is because the classification is done in *P* and *Q* channel in parallel, where the feature dimension in each is around 50% less than in RGB.

### C. Confusion Sets Ranking

For a 10-class classification problem, we may have at most 45 confusion sets for one-versus-one competition. Images with the same top-2 candidates of classes in the baseline decision form one confusion set. The image number in each confusion set varies. For the second stage classification, we give a higher priority to confusion sets that have a large number of images and plot the test accuracy as a function of the total resolved confusing sets in Fig. 5. The test accuracy increases as more confusion sets are resolved and saturates after 27 sets are taken care of.

### D. Effects of Soft-Label Smoothing (SLS)

To demonstrate the power of SLS, we study the binary classification problem and show the results in Table III for

TABLE III  
COMPARISON OF IMAGE-LEVEL TEST ACCURACY (%) FOR FOUR CONFUSION SETS WITH AND WITHOUT SLS.

Confusing Group	intra-hop only	intra-hop and SLS
Cat vs Dog	76.45	79.1
Airplane vs Ship	92.1	93.75
Automobile vs Truck	89.8	92.95
Deer vs Horse	87.8	90.95

four frequent confusion sets: Cat vs Dog, Airplane vs Ship, Automobiles vs Truck, and Deer vs Horse. The training set includes 5,000 images while the test set contains 1,000 images from each of the two classes. For pixel-based classification, we examine test accuracies of intra-hop prediction and the addition of SLS, where results of *P* and *Q* channels are ensembled. We see from the table that the addition of SLS outperforms that without SLS by 1.65%~3.15% in all four confusing sets. The amount of improvement in different confusion sets vary because of their different characteristics and hardness of the objects.

To understand the behavior of SLS furthermore, we show intermediate label maps in heat maps from hop-2 to hop-4 for three Dog images in Fig. 6 with three schemes: a) intra-hop prediction, b) one iteration of SLS label initialization only and c) SLS with another iteration of cross-hop label update. The heatmaps indicate the confidence score at each pixel location for the Dog class. For the hop-2 heatmaps, regions composed by the dog face have higher Dog confidence than other regions. By comparing Figs. 6(a) and (b), we see an increase in the highest confidence level in hop-3 using one round of SLS label initialization. Clearly, SLS makes label maps more distinguishable. Furthermore, heatmaps are more consistent across hops in Fig. 6(b). After one more SLS iteration of cross-hop update, the confidence scores become more homogeneous in a neighborhood as shown in Fig. 6(c), where more locations agree with each other and with the ground truth label.

Since SLS is an iterative process, we show the pixel-level accuracy and the final image-level accuracy as a function of the iteration number for the four confusion sets discussed earlier in Fig. 7. The first point is the initial label in SLS. We see that the pixel-level accuracy at each hop increases rapidly in first several iterations. This is especially obvious in shallow hops. Then, these curves converge at a certain level. In the experiments, we set the maximum number of iterations to  $num\_iter = 3$  for the one-versus-one confusion set resolution to avoid over-smoothing. In the 10-class baseline, we set  $num\_iter = 1$  (i.e. only the initialization of SLS) by taking both inter-hop guidance and model complexity into account.

Although the pixel-level accuracy saturates at the same level for all three hops, the ensemble performance (in red dashed line) is better than that of each individual hop as shown in Fig. 7. As the iteration number increases, the ensemble result also increases and saturates in 3 to 5 iterations although the increment is smaller than that of the pixel-level accuracy.

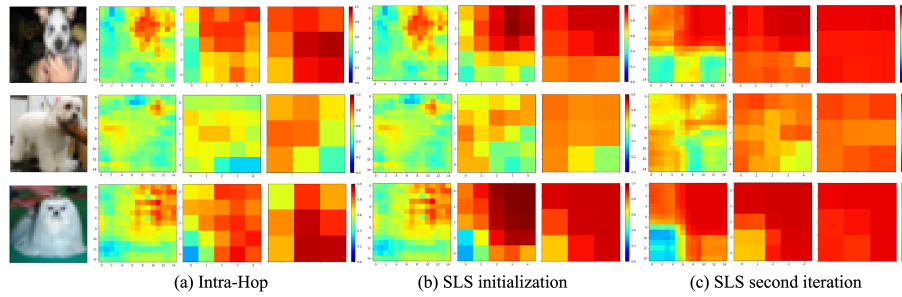


Fig. 6. Illustration of intermediate label maps for different methods, where each subfigure shows label heatmaps in hop-2, hop-3 and hop-4 (from left to right).

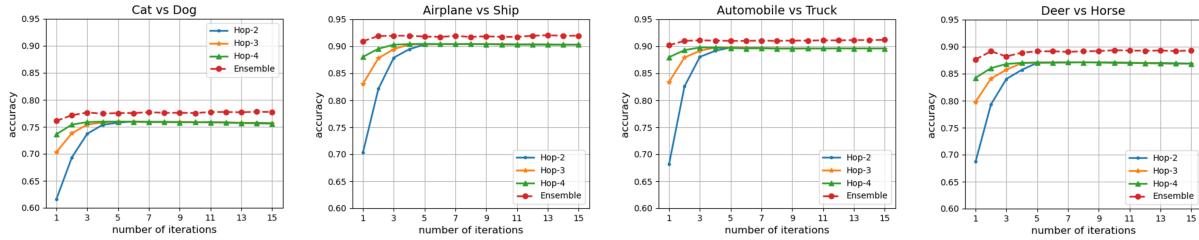


Fig. 7. Pixel-level test accuracy in hop-2, hop-3 and hop-4 and the image-level ensemble as a function of iteration numbers for P-channel images only.

### E. Ablation Study and Performance Benchmarking

The ablation study of adopting various components in E-PixelHop is summarized in Table IV, where the image-level test accuracy for CIFAR-10 is given in the last column. The study includes: data augmentation, color channel decomposition, the methods of pixel-level classification in the baseline and in confusion set resolution, respectively. The first four rows show the baseline performance without confusion set resolution while the last three rows have both. For pixel-level classification, the intra-hop column does not have label smoothing while the SLS\_init column has SLS initialization only. Moreover, the pixel-level classification in the baseline does not use augmented images to train classifiers by considering the time and model complexity. Data augmentation is only used for the meta classification in the baseline and one-versus-one competition in confusion set resolution. The E-PixelHop baseline achieves test accuracy of 73.72% using both P/Q channels with SLS initialization for pixel-level classification and data augmentation when training the meta classifier. By further adding the 2<sup>nd</sup> pipeline with three SLS iterations in pixel-level classification, E-PixelHop achieves a test accuracy of 76.18% as shown in the sixth row, where all 45 confusing sets are processed.

We compare the performance of six methods in Table V. They are modified LeNet-5 [1], PixelHop [6], PixelHop<sup>+</sup> [6] and PixelHop++ [7], the E-PixelHop baseline, and the complete E-PixelHop. To handle color images, we modify the LeNet-5 network architectures slightly, whose hyper parameters are given in Table VI. Both the E-PixelHop baseline and the complete E-PixelHop outperform other benchmarking methods. An improvement of 9.37% over PixelHop++ and an improvement of 3.52% over PixelHop<sup>+</sup> are observed, respectively.

### VI. CONCLUSION AND FUTURE WORK

An enhanced SSL-based object classification method called E-PixelHop was proposed in this work. It has two classification stages: multi-class classification in the first stage and the confusion set resolution in the second stage. Soft-label smoothing (SLS) was proposed to enhance the performance and ensure consistency of intra-hop and inter-hop pixel-level predictions. Effectiveness of SLS shows the importance of prediction agreement at different scales. Furthermore, it is important to handle confusing classes carefully so that the overall classification performance can be boosted.

For the second-stage classification, all training images in the confusion set are used to train the one-versus-one model in the current E-PixelHop. In the future, it is worthwhile to investigate boosting with hard cases mining and learn from errors. Furthermore, not every pixel in an image is discriminant and salient. We should find a way to focus on discriminant regions by introducing an attention mechanism. It is also desired to generalize our work to the recognition of a larger number of classes (e.g., CIFAR-100) and higher image resolution (e.g., the ImageNet).

### REFERENCES

- [1] 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.
- [2] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
- [3] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.
- [4] 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*, pp. 770–778, 2016.

TABLE IV  
ABLATION STUDY OF E-PIXELHOP'S COMPONENTS FOR CIFAR-10

Augmentation	Color Channels		Baseline		2-nd pipeline		Test Accuracy
	P channel	Q channel	intra-hop	SLS_init	intra-Hop	SLS	
	✓	✓		✓			68.69
✓	✓			✓			70.26
✓	✓	✓		✓			<b>73.72</b>
✓	✓		✓				72.29
✓	✓	✓		✓	✓		72.79
✓	✓	✓		✓		✓	<b>76.18</b>
✓	✓	✓	✓			✓	75.54

TABLE V  
COMPARISON OF TESTING ACCURACY (%) WITH LeNET-5, PIXELHOP, PIXELHOP+ AND PIXELHOP++ FOR CIFAR-10.

	Test Accuracy (%)
Lenet-5	68.72
PixelHop [6]	71.37
PixelHop+ [6]	72.66
PixelHop++ [7]	66.81
E-PixelHop Baseline (Ours)	73.72
E-PixelHop (Ours)	<b>76.18</b>

TABLE VI  
HYPER PARAMETERS OF THE MODIFIED LeNET-5 NETWORK AS COMPARED WITH THOSE OF THE ORIGINAL LeNET-5.

	Original LeNet-5	Modified LeNet-5
Conv-1 Kernel Size	5x5x1	5x5x3
Conv-1 Kernel No.	6	32
Conv-2 Kernel Size	5x5x6	5x5x32
Conv-2 Kernel No.	16	64
FC-1	120	200
FC-2	84	100
Output Layer	10	10

- fashion image classification,” *Expert systems with applications*, vol. 116, pp. 328–339, 2019.
- [16] A. Zweig and D. Weinshall, “Exploiting object hierarchy: Combining models from different category levels,” in *2007 IEEE 11th International Conference on Computer Vision*, pp. 1–8, IEEE, 2007.
- [17] A. Bilal, A. Jourabloo, M. Ye, X. Liu, and L. Ren, “Do convolutional neural networks learn class hierarchy?,” *IEEE transactions on visualization and computer graphics*, vol. 24, no. 1, pp. 152–162, 2017.
- [18] Z. Yan, H. Zhang, R. Piramuthu, V. Jagadeesh, D. DeCoste, W. Di, and Y. Yu, “Hd-cnn: hierarchical deep convolutional neural networks for large scale visual recognition,” in *Proceedings of the IEEE international conference on computer vision*, pp. 2740–2748, 2015.
- [19] C. Chen, S. Li, X. Fu, Y. Ren, Y. Chen, and C.-C. J. Kuo, “Exploring confusing scene classes for the places dataset: Insights and solutions,” in *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pp. 550–558, IEEE, 2017.
- [20] S. Li, C. Chen, Y. Ren, and C.-C. J. Kuo, “Improving object classification performance via confusing categories study,” in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1774–1783, IEEE, 2018.
- [21] X. Zhu and M. Bain, “B-cnn: branch convolutional neural network for hierarchical classification,” *arXiv preprint arXiv:1709.09890*, 2017.
- [22] 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.
- [23] C.-C. J. Kuo, “The cnn as a guided multilayer recos transform [lecture notes],” *IEEE signal processing magazine*, vol. 34, no. 3, pp. 81–89, 2017.
- [24] 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.
- [25] 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*, 2019.
- [26] Y. Chen, Z. Xu, S. Cai, Y. Lang, and C.-C. J. Kuo, “A saak transform approach to efficient, scalable and robust handwritten digits recognition,” in *2018 Picture Coding Symposium (PCS)*, pp. 174–178, IEEE, 2018.
- [27] Y. Chen, Y. Yang, W. Wang, and C.-C. J. Kuo, “Ensembles of feedforward-designed convolutional neural networks,” in *2019 IEEE International Conference on Image Processing (ICIP)*, pp. 3796–3800, IEEE, 2019.
- [28] 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.
- [29] M. Zhang, Y. Wang, P. Kadam, S. Liu, and C.-C. J. Kuo, “Pointhop++: A lightweight learning model on point sets for 3d classification,” in *2020 IEEE International Conference on Image Processing (ICIP)*, pp. 3319–3323, IEEE, 2020.
- [30] 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,” *arXiv preprint arXiv:2007.09510*, 2020.
- [31] X. Liu, F. Xing, C. Yang, C.-C. J. Kuo, S. Babu, G. E. Fakhri, T. Jenkins, and J. Woo, “Voxelhop: Successive subspace learning for als disease classification using structural mri,” *arXiv preprint arXiv:2101.05131*, 2021.
- [32] A. Krizhevsky and G. Hinton, “Learning multiple layers of features from tiny images,” tech. rep., University of Toronto, Toronto, Ontario, 2009.
- [33] T. Xie, B. Wang, and C.-C. J. Kuo, “Graphhop: An enhanced label propagation method for node classification,” *arXiv preprint arXiv:2101.02326*, 2021.
- [5] 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*, pp. 4700–4708, 2017.
- [6] 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.
- [7] 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)*, pp. 3294–3298, IEEE, 2020.
- [8] J. Mutch and D. G. Lowe, “Multiclass object recognition with sparse, localized features,” in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’06)*, vol. 1, pp. 11–18, IEEE, 2006.
- [9] P. Schnitzspan, M. Fritz, and B. Schiele, “Hierarchical support vector random fields: Joint training to combine local and global features,” in *European conference on computer vision*, pp. 527–540, Springer, 2008.
- [10] K. Fukushima and S. Miyake, “Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition,” in *Competition and cooperation in neural nets*, pp. 267–285, Springer, 1982.
- [11] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio, “Robust object recognition with cortex-like mechanisms,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 29, no. 3, pp. 411–426, 2007.
- [12] L. Liu, C. Shen, and A. van den Hengel, “The treasure beneath convolutional layers: Cross-convolutional-layer pooling for image classification,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4749–4757, 2015.
- [13] S. Jetley, N. A. Lord, N. Lee, and P. H. Torr, “Learn to pay attention,” *arXiv preprint arXiv:1804.02391*, 2018.
- [14] W. Liu, I. W. Tsang, and K.-R. Müller, “An easy-to-hard learning paradigm for multiple classes and multiple labels,” *The Journal of Machine Learning Research*, vol. 18, 2017.
- [15] Y. Seo and K.-s. Shin, “Hierarchical convolutional neural networks for