Compact Deep Convolutional Neural Networks for Image Classification

Zejia Zheng, Zhu Li, Abhishek Nagar[1] and Woosung Kang[2]

***Abstract*—Convolutional Neural Network is efficient in learning hierarchical features from large datasets, but its model complexity and large memory foot prints are preventing it from being deployed to devices without a server backend support. Modern CNNs are always trained on GPUs or even GPU clusters with high speed computation power due to the immense size of the network. Methods on regulating the size of the network, on the other hand, are rarely studied. In this paper we present a novel compact architecture that minimizes the number of lower level kernels in a CNN by separating the color information from the original image. A 9-patch histogram extractor is built to exploit the separated color information. A higher level classifier learns the combined features from the compact CNN, trained only on grayscale image with limited number of kernels, and the histogram extractor. We apply our compact architecture to CIFAR-10 and Samsung Mobile Image Dataset. The proposed architecture has a recognition accuracy on par with those of state of the art CNNs, with 40% less parameters.**

I. Introduction

Convolutional Neural Network (CNN) [7] is one of the leading image classification architectures for hierarchical feature extraction. CNNs have been reported to have state of the art performance on many image recognition and classification tasks, including hand written digit recognition [6], house numbers recognition [10], traffic signs classification [2], and 1000 class ImageNet dataset classification and localization [5], [11].

Despite these encouraging progresses, there is still limited research on compact convolutional neural networks that can be easily implemented on to a mobile device. The large amount of parameters inside current state of the art CNNs makes it hard for mobile devices to label an arbitrary RGB image in short time. In this paper we propose a CNN based architecture that uses minimal number of lower level kernels while maintaining the high performance of a CNN with more parameters in lower level layers. The network is trained only on grayscale images thus both the size and the number of the kernel on 1st layer can be reduced. This leads to a 40% drop on the final size of the network. The loss in the capability of the network introduced by limiting the lower level feature extractor is amended with the help of a carefully crafted color histogram feature vector extracted from patches of the original image. Several different configurations of the combination are tested.

We report the experiment result on CIFAR-10 and Samsung Mobile Image Dataset. The CIFAR-10 dataset has been heavily tested on by previous works [4] [5] [8]. Our result shows that the compact architecture achieves similar performance with minimal number of parameters (40% less). The Samsung Mobile Image Dataset is a hierarchical dataset with more than 80000 images and 31 class labels. The images have higher resolution compared to the CIFAR-10 dataset, which makes the histogram feature vector more useful. As a result combining hand crafted histogram feature vector with the CNN final feature vector improves the accuracy of the CNN classifier (on grayscale images) by 4%, achieving same performance compared to a single CNN trained on RGB images. The final architecture is much more compact compared to the original version, while the performance is similar, sometimes better, compared to a single CNN trained on RGB images.

Our contributions in this paper can be summarized as follows:

- We propose a compact architecture based on the combination of CNN and hand-crafted color histogram feature extractor. The proposed architecture minimizes network size by separating color information from the original image, thus limiting the number of kernels required to extract feature from the grayscale input. The compact network has 40% less parameter to tune with but it maintains the performance of the original CNN trained on RGB images.
- We apply our compact network to a hierarchical dataset (i.e. Samsung Mobile Image Dataset) with clean basic categories and confusing subcategories. The experiment result reveals that hand crafted feature (i.e. 9 patch color histogram) helps the network to clarify the boundaries among classes in the same basic category. Global and local histogram vector is more useful when the image contains more information (i.e. high resolution).

II. Related Work

A. Convolutional Neural Network

In recent years commercial and academic datasets for image classification have been growing at an unprecedented pace. The SUN database for scenery classification contains 899 categories and 130,519 images [14]. The ImageNet dataset contains 1000 categories and 1.2 million images [5]. In response to this immensely increased complexity, a great many researchers have focused on increasing the depth of classifiers to capture invariance and useful features.

Among a great number of available deep architectures, Convolutional Neural Network (CNN) is reported to have the leading performance on many image classification tasks.

[1] Zejia Zheng, Zhu Li, Abhisheck Nagar are with Samsung Research America's Multimedia Core Standards Research Lab in Richardson, TX. Zejia Zheng is also a Ph.D. student studying at Michigan State University, EI Lab.

[2] Woosung Kang is with Samsung Electronics, Soul, Korea.

Overfeat, a CNN-based image features extractor and classifier, scored a 29.8% error rate in classification and localization task on ImageNet 2013 dataset. Clarifai, a hierarchical architecture of CNN and deconvlutional neural network, achieved an 11.19% error recognition rate on ImageNet 2013 classification task [15].

It has also been reported that the performance of CNN is highly correlated with the number of layers. Winners of the competitions mentioned above have millions of parameters to tune with, which requires a large number of training samples. The ILSVRC 2012 challenge winner CNN by Krizhevsky has around 60 million parameters [5]. Overfeat, the ILSVRC 2013 challenge winning CNN, has more than 140 million parameters [11]. These networks are always trained on a GPU machine or GPU clusters for better performance.

As is introduced in previous section, our goal in this paper is to find a compact architecture that balances the size of the network and its performance on device based image classification task. Thus we do not attempt to outperform the existing works in [5], [11] on CIFAR-10.

B. Histogram-based Classification

Color histograms are widely used to compare images despite the simplicity of this method. It has been proven to have good performance on image indexing with relatively small datasets [12]. Color histograms are trivial to compute and tend to be robust against small changes to camera viewpoint, which makes them a good compact image descriptor. It was also reported in [1] that the performance of a histogram based classifier was improved when the higher level classifier was a support vector machine.

However, when applied to large dataset, histogram based classifiers tend to give poor performance because of high variances within the same category. It is also observed that images with different labels may share similar histograms [9].

In this work, we propose a novel architecture that combines the histogram-based classification method with CNN. The histogram representation of color information helps the CNN to exploit color information in the original image. This means that we can minimize the size of the basic feature detectors (i.e. layer 1 of the CNN). The proposed architecture is introduced in the following section.

III. Compact CNN with Color Descriptor

A. Deep Convolutional Neural Networks

We use the architecture of Krizhevsky et al. [5] to train the ‘original’ CNN in the experiments. We then modified layer 1 by changing the kernel size (from $5 \times 5 \times 3$ to $5 \times 5 \times 1$) and the number of kernels (from 64 to 32) in later experiments. The details of the experiments are introduced in the next section.

We trained two CNNs with different number of kernels in the first layer: an original version and a compact version. The ‘original’ network is the exact replicate of the CNN reported in [4], which gave a final error recognition rate of 13% using multi-view testing. In this work, however, we only use single view testing when reporting the final result for both the original CNN and compact CNN.

Both the original and the compact CNNs have four convolution layers. Table I shows the details of the two networks when trained on cropped images from the CIFAR-10 dataset. Our compact CNN is marked in bold font to show the difference. There are only 32 kernels in the first layer of the compact CNN while the number is 64 in the original CNN. This cuts down the number of parameters by 50% in layer 3 (i.e. the 2nd convolution layer), thus the final compact CNN has 40% less parameters to tune compared to the original version.

The convolution operation is expressed as:

$$y^{j(r)} = ReLU(b^{j(r)} + \sum_i k^{ij(r)} * x^{i(r)}) \tag{1}$$

where x^i is the ith input map and y^j is the jth output map. k^{ij} is the convolution kernel corresponding to the ith input map and the jth output map. r indicates a local region on the input map where the weights are shared.

ReLU non-linearity (i.e. $ReLU(x) = max(0, x)$) is used in the network. It is observed that ReLU yields better performance and faster convergence speed when trained by error back propagation [5].

Max pooling is done in a 3×3 sliding window at a 2×2 stride size in layer 2 and layer 4. This helps the network to extract the most prominent low level features and reduce the size of feature vector.

More details about the experiment set up can be found in Fig. 1 and table I.

TABLE II
Size of different configuration

CNN configuration	Total num of parameter
original grayscale $k = 1$	143168
original RGB $k = 3$	146368
Compact Architecture	**91168**

B. Color Information

A color is represented by a three dimensional vector corresponding to a point in the color space. We choose red-green-blue (RGB) as our color space, which is in bijection with the hue-saturation-value (HSV).

HSV may seem attractive in theory for a classifier purely based on histograms. HSV color space separates color component from the luminance component, making the histogram less sensitive to illumination changes. However, this does not seem to be important in practice. [1] reports minimal improvement when switching from RGB color space to HSV color space.

The choice for the choice of RGB is that the three channels share the same range (i.e. from 0-255), making it easier for normalization.

We experiment with three different configurations of the color histogram:

TABLE I
ORIGINAL AND COMPACT CNN ARCHITECTURE (CIFAR-10)

	layer 1	layer 2	layer 3	layer 4	layer 5	layer 6	layer 7	layer 8
operation	conv	max	conv	max	conv	conv	fully connect	softmax
original input size	24x24xk	24x24x64	12x12x64	12x12x64	6x6x64	6x6x32	6x6x32	10x1
compact input size	**24x24x1**	**24x24x32**	**12x12x32**	**12x12x64**	**6x6x64**	**6x6x32**	**6x6x32**	**10x1**
filter size	5x5xk		5x5x64		3x3x64	3x3x32	6x6x32	
compact filter size	**5x5x1**		**5x5x32**		**3x3x64**	**3x3x32**	**6x6x32**	
original filter num	64		64		32	32	10	
compact filter num	**32**		**64**		**32**	**32**	**10**	
pool size		3x3		3x3				
stride	1x1	2x2	1x1	2x2	1x1	1x1		
output	24x24x64	12x12x64	12x12x64	6x6x64	6x6x32	6x6x32	10x1	

1) Global histogram, 48 bins. In this setup we examine if global color information helps with the classification.
2) 9-patch histogram, 192 bins. The 9 patches are generated as is shown in Fig. 1. As CIFAR-10 dataset contains only 32 by 32 images, which makes it harder to extract useful histograms, the number of bins in this setup should be 48, 2×24, 2×24, and 4×24.
3) 9-patch histogram, 384 bins. Numbers of bins are doubled compared to the previous set up.

These experiments on histogram configuration are solely carried out on the CIFAR image dataset. This series of experiment serves as a guideline for our experiment on Samsung Mobile Image Dataset.

C. Combined Architecture

Once the CNN is trained for the classification task with the grayscale version of the training set, we replace the fully connected layer and the softmax layer (i.e. layer 7 and 8 as is shown in table I) with a new fully connected layer and a new softmax layer trained on the combined feature vector, using the feature vector from the same training set.

The combined feature vector is generated by algorithm 1.

```
**Input**: image $I$, total patch number $k$
**Output**: Combined Feature Vector $vec\_combined$
segment $I$ into $\{I_i, i = 1, 2, ..., k\}$;
extract histogram vector $hist\_vec$ from $\{I_i\}$;
resize $I$ to CNN input size, feed I into CNN;
extract layer 6 output $cnn\_layer\_6\_vec$ from CNN;
reshape $cnn\_layer\_6\_vec$ to a one dimensional vector $cnn\_vec$;
$vec\_combined = concatenate(cnn\_vec, hist\_vec)$;
**return** $vec\_combined$
```

Algorithm 1: EXTRACT NEW FEATURE VECTOR

With the new feature vector extracted from both the training set and testing set, we train a new layer 7 (fully connected layer) and layer 8 (softmax layer) based on the combined feature vector extracted from the training set.

The purpose of this work is to find a compact architecture by combining handcrafted feature representation with final feature vector from the CNN. To make clear comparison, we evaluate the performance of the combined classifier with several different setups:

1) Cropped images and uncropped images. Training on cropped images means that we feed patches of image into the network instead of the original image. This allows the network to train with relatively more samples, but would jeopardize recognition for certain classes in Samsung Mobile Image Dataset (e.g. upper body and whole body).
2) Colored images and grayscale images. We use recognition on uncropped color images as the base line for performance evaluation. The propose compact architecture, however, separates color information from the original image, and feed only grayscale image to the pretrained CNN.
3) CIFAR-10 dataset and Samsung Mobile Image Dataset. We use the CIFAR-10 dataset to test different configurations of histograms and several data augmentation methods. The results on CIFAR-10 serves as a guideline for us to construct a compact classifier for the Samsung Mobile Image Dataset, a hierarchical dataset collected at Samsung Research America.

Details about these experiments are reported in the following section. In short, we found that the proposed compact architecture trained on cropped grayscale image maintains the high accuracy of the original CNN trained on cropped RGB images.

IV. EXPERIMENT

A. CIFAR dataset

The CIFAR-10 dataset consists of 60000 32 by 32 color images from 10 basic categories. The class labels are: airplane, automobile, bird, cat, deer, dog, frog, horse, ship and truck. There are 6000 images per class, with 50000 training images and 10000 test images. The image included in the dataset is assumed to be easy to named by a human classifier without ambiguity. The dataset is collected by Krizhevsky and Hinton and is reported in [4].

CIFAR-10 has been heavily tested on with many classification methods. Krivzhevsky et al. [5] achieved a 13% test error rate when using their ILSVRC 2012 winning CNN architecture (without normalization). By generalizing Hinton's dropout [3] into suppression in weight values instead of activation values, Wan et al. [13] reported a error testing rate of 9.32 %, using their modified Convolutional Neural Network DropConnect. Lin et al. [8] replaced the ReLU convolutional layer in

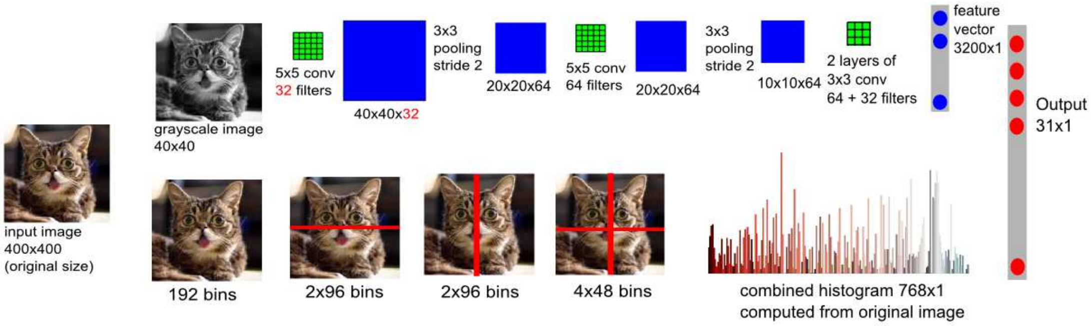

Fig. 1. Compact CNN with histogram based color descriptor. We separate color information from the original image by only feeding the CNN with the grayscale image. Color histogram is combined with the final feature vector. This figure shows how an image from Samsung Mobile Image Dataset is classified as is described in section IV-B. Image size and the number of bins in a histogram are reduced accordingly when testing on CIFAR-10. There are only 32 filters in layer 1, compared to 64 filters in layer 1 of the original network. The performance of the Compact architecture, however, is similar to the original architecture, with the network size 40% smaller when testing on CIFAR-10, and 20% smaller when testing on Samsung Mobile Image Database.

Krivzhevsky's architecture [5] with a convolutional multi-layer perceptron. They reported a test error rate of 8.8 %, currently ranking top on the leader board of classification on CIFAR-10 dataset.

Despite the improvement and variations described above, our experiment in this work is still based on Krizhevsky's architecture as is described in [5]. The goal of this paper is to study the contribution of color information to CNN based image classification, and to seek possible combination between hand crafted feature vector and CNN extracted feature vector to further exploit the low level features with limited number of parameters. For these reasons we apply our modifications to a standard CNN architecture as is provided by Krizhevsky in [5]. We believe that the combined architecture can also be applied to other CNN variants with few modifications.

1) Getting Histogram: Because CIFAR consists of images with only 1024 pixels, getting a large histogram vector would be meaningless. Therefore we only extract a global histogram of 48 bins from the original image in our first experiment. The histogram and the final feature vector from the CNN pass are concatenated together as is described in algorithm 1.

In later trials, we move on to more complicated histograms feature vector extraction configurations instead of just using the global histogram. We extracted histogram feature vectors of different length from 9 patches of the input image. Suppose we are to extract a histogram feature vector of length 192, then the number of bins of each patch would be: 48 bins from the entire image, 24×4 bins from the left half, the right half, the top half and the bottom half, 12×4 bins from the upper left corner, the upper right corner, the lower left corner and the lower right corner. The intention is to reflect the global color information as well as the local color distribution at certain precision to exploit the color details.

2) Training Methods: Although our CNN architecture is similar to Krivzhevsky's network, we modify some parts of the training procedures in [5] to suit our needs.

When trained on CIFAR-10 dataset, the first few CNNs are not trained on cropped images as is described in [5]. By training the network on five image patches (top left, top right, lower left, lower right, and center) and their horizontal flip, Krivzhevsky was able to enlarge the size of the training dataset and generate more robust representations inside the network. We do not use cropped CIFAR images on initial trials for the following reasons: (1) our intention in this work is to evaluate the effectiveness of directly feeding the classification layer with hand-crafted histogram instead of relying on the CNN to exploit color information. The comparison would be more straightforward when we use the entire image instead of image patches. (2) cropping data may not be the best idea in certain applications. In Samsung Mobile Image Dataset, for example, the classifier needs to distinguish human upperbody from the entire human body. Training on certain patches may introduce confusion. (3) performance issue. Training on the entire image instead of image patches reduces training time. We report results on cropped images in later experiments, as is shown in table III.

Another difference between our network and Krizhevsky's reported 13% error recognition CNN is that we do not report result based on multiview tests. By adopting multiview test instead of single shot test, the 13% error CNN takes patches of images (and their horizontal flips) as input and aggregates the final output probability. As our intention is to improve the network architecture, we feel that comparison should be done with single shot tests. Our work, however, can be easily generalized to multiview testing scenarios. Performance is expected to be improved accordingly.

We use mini-batches of 128 examples, momentum of 0.9 and weight decay of 0.004. All networks are initialized with learning rates of 0.001. The learning rates are manually adjusted (lowered by a factor of 10) whenever the validation error stops improving (3 times at most).

3) Experiment Result: Experiment results on CIFAR-10 are listed in table III and table IV.

In table III, we first explore the configuration of histogram vector by adjusting the amount of information the histogram vector contains. In each case, the grayscale CNN, trained on the original architecture remains unchanged. Although global color histogram does not help to improve classification, the 9-patch configuration boosted the performance as expected. One important guideline we find out is that a more detailed histogram (384 bins) gives better classification result compared to rough color information.

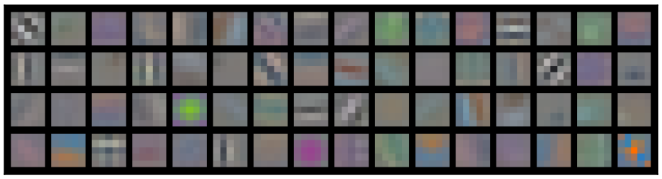

Fig. 2. Compact CNN layer 1 kernel. There are only 32 kernels in layer 1 of the proposed architecture. The network learns basic features as edges and corners from the grayscale input. Network trained on grayscale images from Samsung Mobile Image Dataset.

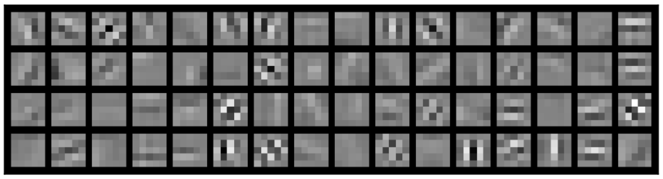

Fig. 3. Original CNN layer 1 kernel. There are 64 kernels in layer 1 of the original architecture. The network learns the basic features as in the compact architecture, but there are ’dead’ kernels as is observed in [11].Network trained on grayscale images from Samsung Mobile Image Dataset.

When trained on uncropped RGB images using the original architecture, the performance (recognition rate) is 2% worse than the original architecture trained on grayscale images. This can be explained by the following two reasons: (1) looking back to the dataset, distinctions among classes rely heavily on the shape rather than the color information of the class. With same amount of lower level kernels, CNN trained on grayscale images can focus the more of the resource on the shape (i.e. corners and edges)while the CNN trained on rgb images needs to distinct edges and corners with different colors. (2) when trained on uncropped images, there is not enough images (5 times less compared to the CNN trained on cropped images) for the network to distinguish higher dimensional features ($32 \times 32 \times 3$ in the case of RGB images).

When trained with enough images (i.e. after cropping), the CNN trained with RGB images is more accurate, with an error recognition rate of 16.36%. The accuracy can be further lowered to 13.10% when using multiview testing. However, the original CNN has 146368 parameters, due to the large number of kernels in layer 1 and layer 2. The compact CNN trained on grayscale images has less filters in layer 1 (50% compared to the original CNN), while the error recognition rate rises only by 1%. As a result, the proposed architecture maintains high performance of the while the size of the architecture is 40% smaller.

The improvement can be explained as follows: although there are fewer kernels in the first layer, these kernels are extracting information from a lower dimension space compared to the original version.

B. Samsung Mobile Image Dataset

The Samsung Mobile Image Dataset is an industrial dataset collected at Samsung. There are 31 classes, with a total 82181 images of different sizes and resolutions.

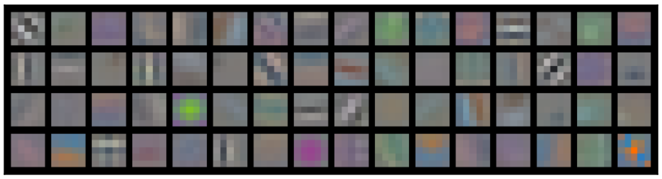

Fig. 4. Original CNN trained on RGB images from Samsung Mobile Image Dataset. The network deploys most of its resources in finding color gradient, compared to the kernels learned in CNN trained on grayscale images.

TABLE III
CIFAR-10 RESULT ON UNCROPPED IMAGES USING ORIGINAL CNN

Input image and hist config	Test error rate
grayscale	24.79%
grayscale+global hist (48 bins)	24.95%
grayscale+9 patch hist (192 bins)	24.55%
grayscale+9 patch hist (384 bins)	24.10%
rgb	27.12%
rgb + 9 patch hist (384 bins)	26.95%

The detailed number of images per class is listed in the Appendix.

Class names together with sample images of each class are shown in Fig. 5. It can be seen from table V that the boundaries among classes in this dataset are much more unclear compared to the CIFAR dataset. For example, we are not only training the network to recognize that a person is in the image, we are also requiring the network to report a general posture (e.g. lying, leaning forward or backward, etc.). The general food category is also divided into three sub categories: the class ‘food part 1’ contains breads, desserts and bottled/cupped food; the class ‘food part 2’ contains meat and other foods on a plate; the class ‘food part 3’ consists of pictures about foods on tables. Detailed clarification of each class can be found in table VI. This dataset is a hierarchical dataset which makes it easy to get the basic labels correct but hard to distinguish classes within the same general category.

We split the dataset by assigning 10% of the images to the testing set, 10% to a validation set and 70% to the training set. After the 768 bins histogram is extracted, each image is then resized and cropped into a 48×48 grayscale image and then fed to the convolution network. The layer configuration and parameters are the same as is described in table I. Note that the input image size should be modified accordingly.

1) Getting Histogram: As the original image contains more detailed information due to the increased image resolution, a histogram vector of length 384 is not sufficient to describe the color information with high accuracy.

Guided by the result from our first experiment on CIFAR dataset, we extract a color descriptor of length 768 by concatenating histogram feature vectors from 9 patches of the image as is described in previous experiment.

2) Data Augmentation: As is reported in the previous experiment, cropping images leads to more robust features learned by the network. But cropping as is done in [5] may lead to confusion when the network needs to distinguish upper body from whole body (class 9 and 10 in table VI). Therefore we take a more careful cropping process by only flipping

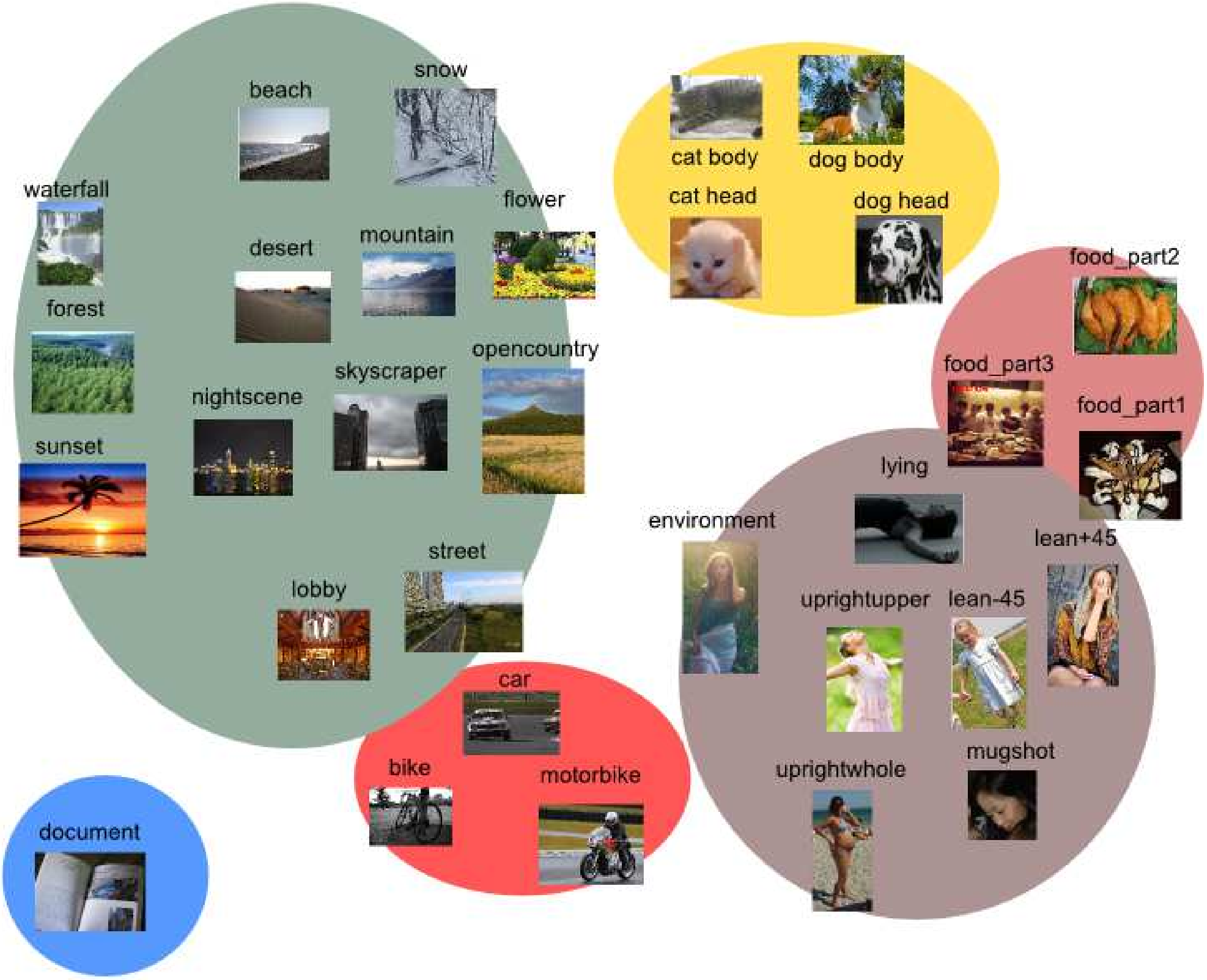

Fig. 5. Sample images for Samsung Mobile Image Dataset. This hierarchical image dataset has unclear boundaries among categories. The first level category is presented by colored ovals. Second level categories are presented by the label and a random sample from the training dataset. The images are of higher resolution and more confusing compared to the CIFAR-10 dataset.

TABLE IV
CIFAR-10 TEST RESULT

Architecture (all on cropped images)	Test error rate	Number of parameters
grayscale (original)	18.10%	143168
grayscale (original) 9 patch hist (384 bins)	17.55%	147008
grayscale (compact)	18.95%	91168
grayscale (compact) 9 patch hist (384 bins)	**17.64%**	**95008**
rgb (original)	16.36%	146368

images from the uprightwhole class horizontally at a 50% probability. The images are then resized and zero-padded to fit the input size of the network (40×40).

3) Experiment Result: The error recognition rates of different configurations are reported in table V.

The difference between the error recognition rate of the original architecture (trained on grayscale images) and the compact architecture (trained on grayscale images) is even smaller when using Samsung Mobile Image Dataset (i.e. less than 0.3%). This result indicates that the 64 filters on the first layer learned redundant information. The learned filters are visualized in Fig. 2, Fig. 3, and Fig. 4.

It can also be seen from the result that color information boosts the performance grayscale CNN (original version and compact version) by as much as 3% (for compact CNN) and 4% (for original CNN). Our proposed architecture is neck and neck with the original architecture in recognition, while the proposed architecture is 20% more compact compared to the original version. There are more minimization in previous experiment (40% vs. 20%) because we are now training on larger input sizes, with much more weight on the fully connected layers connecting the final feature vector with the softmax layer.

TABLE V
SAMSUNG MOBILE IMAGE TEST RESULT

Architecture (all on cropped images)	Test error rate	Number of parameters
grayscale (original)	26.08%	230848
grayscale (original) 9 patch hist (798 bins)	22.80%	238528
grayscale (compact)	26.06%	178848
grayscale (compact) 9 patch hist (789 bins)	**22.99%**	**186528**
rgb (original)	22.61%	234048

The experiment result shows that histogram vectors are more helpful when the training image is of higher resolution.

V. CONCLUSIONS

In this paper we present a novel compact architecture for image classification. The proposed architecture combines hand-crafted color information with a convolutional neural network pre-trained with thumbnail grayscale images. The proposed architecture has similar recognition capacity compared to state of the art CNNs but with a much smaller network size (40% less when testing on CIFAR-10). We apply our network to CIFAR-10 dataset, a standard image classification benchmark, and Samsung Mobile Image Dataset, a hierarchical image dataset. The experiment shows that carefully designed histogram extractor helps to boost the performance of the convolutional neural network, and the boost is much more significant when the image is of higher resolution.

APPENDIX

Details about the Samsung Mobile Image dataset are included in table VI.

TABLE VI
CLASS LABELS AND NUMBER OF IMAGES PER CLASS

Level 1	Level 2	# of Images
vehicle	bike	3097
	motorbike	865
	car	2969
people	environment	2713
	lean-45	1271
	lean+45	1277
	lying	1005
	mugshot	3625
	uprightupper	4197
	uprightwhole	3336
food	food part1	3291
	food part2	2926
	food part3	3168
documents	document	3080
pets	cat body	3717
	cat head	3521
	dog body	3769
	dog head	3158
flower	flower	3577
scenery	mountain	2838
	skyscraper	2549
	opencountry	1829
	snow	1955
	street	1966
	sunset	2350
	waterfall	1012
	beach	2874
	desert	873
	forest	2667
	lobby	2298
	nightscene	3050

REFERENCES

[1] Olivier Chapelle, Patrick Haffner, and Vladimir N Vapnik. Support vector machines for histogram-based image classification. *Neural Networks, IEEE Transactions on*, 10(5):1055–1064, 1999.

[2] Dan Ciresan, Ueli Meier, and Jürgen Schmidhuber. Multi-column deep neural networks for image classification. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3642–3649. IEEE, 2012.

[3] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

[4] Alex Krizhevsky. Learning multiple layers of features from tiny images. *Unpublisheds*.

[5] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[6] B Boser Le Cun, JS Denker, D Henderson, Richard E Howard, W Hubbard, and Lawrence D Jackel. Handwritten digit recognition with a back-propagation network. In *Advances in neural information processing systems*. Citeseer, 1990.

[7] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[8] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *CoRR*, abs/1312.4400, 2013.

[9] Greg Pass and Ramin Zabih. Histogram refinement for content-based image retrieval. In *Applications of Computer Vision, 1996. WACV'96., Proceedings 3rd IEEE Workshop on*, pages 96–102. IEEE, 1996.

[10] Pierre Sermanet, Soumith Chintala, and Yann LeCun. Convolutional neural networks applied to house numbers digit classification. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 3288–3291. IEEE, 2012.

[11] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013.

[12] Michael J Swain and Dana H Ballard. Indexing via color histograms. In *Active Perception and Robot Vision*, pages 261–273. Springer, 1992.

[13] Li Wan, Matthew Zeiler, Sixin Zhang, Yann L Cun, and Rob Fergus. Regularization of neural networks using dropconnect. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 1058–1066, 2013.

[14] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *Computer vision and pattern recognition (CVPR), 2010 IEEE conference on*, pages 3485–3492. IEEE, 2010.

[15] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional neural networks. *arXiv preprint arXiv:1311.2901*, 2013.