Chapter 8
Compact Deep Neural Networks for Device-Based Image Classification

Zejia Zheng, Zhu Li and Abhishek Nagar

Abstract Convolutional Neural Network (CNN) is efficient in learning hierarchical features from large image datasets, but its model complexity and large memory footprints prevent it from being deployed to devices without a server back-end support. Modern CNNs are always trained on GPUs or even GPU clusters with high-speed computation capability due to the immense size of the network. A device-based deep learning CNN engine for image classification can be very useful for situations where server back end is either not available, or its communication link is weak and unreliable. Methods on regulating the size of the network, on the other hand, are rarely studied. In this chapter we present a novel compact architecture that minimizes the number and complexity of lower level filters in a CNN by separating the color information from the original image. A 9-patch histogram extractor is built to exploit the unused color information. A high-level classifier is then used to learn the features obtained from the compact CNN that was trained only on grayscale image with limited number of filters and the 9-patch histogram extracted from the color information in the image. We apply our compact architecture to Samsung Mobile Image Dataset for image classification. The proposed solution has a recognition accuracy on par with the state-of-the-art CNNs, while achieving significant reduction in model memory footprint. With these advantages, our system is being deployed to the mobile devices.

Z. Zheng (✉)
Michigan State University, 428 South Shaw Lane, Room 3110, East Lansing, MI 48824, USA
e-mail: zhengzej@msu.edu

Z. Li · A. Nagar
Samsung Research America, 1301 E. Lookout Drive, Richardson, TX 75082, USA
e-mail: zhu1.li@samsung.com

A. Nagar
e-mail: a.nagar@samsung.com

G. Hua and X.-S. Hua (eds.), *Mobile Cloud Visual Media Computing*,
DOI 10.1007/978-3-319-24702-1_8

8.1 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 [15]. The ImageNet dataset contains 1000 categories and 1.2 million images [6]. In response to this immensely increased complexity, many researchers have focused on designing even more sophisticated classifiers to effectively capture all the invariant and discriminative features.

Among a great number of available classifiers, Convolutional Neural Network (CNN) is reported to have the leading performance on many image classification tasks. Overfeat, a CNN-based image features extractor and classifier, scored a low 29.8 % error rate in classification and localization task on ImageNet 2013 dataset. Clarifai, a hierarchical architecture of CNN and deconvolutional neural network, achieved an 11.19 % error recognition rate on ImageNet 2013 classification task [16]. CNNs have been reported to have state-of-the-art performance on many other image recognition and classification tasks, including handwritten digit recognition [7], house numbers recognition [11], and traffic signs classification [2].

8.1.1 Network Architecture

Convolutional Neural Network is specifically designed to handle computer vision problems. A typical CNN is presented in Fig. 8.1. It has the following features that differentiate itself from traditional neural networks:

1. Local receptive field. Each neuron in the convolutional layer accepts only a portion of the entire input image. Thus the learned filters only produce the strongest response to a local input pattern, thereby reinforcing the local nature of typical image features.
2. Shared weights. Each neuron in the convolutional layer shares the same set of filters. This architecture ensures that important local features would be detected regardless of their position in the visual field.
3. Subsampling for dimension reduction. Convolutional neural network alternates between the convolutional and pooling layers. Pooling is performed on overlapping or nonoverlapping neighborhoods of the input to reduce the data dimensions and at the same time find the most prominent features.

Combining those three features together, we have the architecture of a typical CNN as is presented in Fig. 8.1.

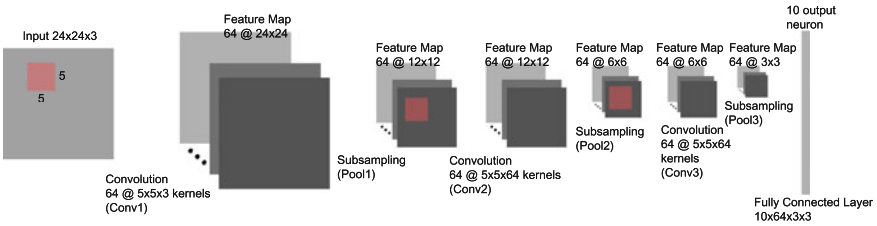

Fig. 8.1 Architecture of a typical CNN. This figure shows the structure of a typical CNN trained on CIFAR-10 dataset

Convolutional Layer

The response map in the convolutional layer is computed using the same set of filters (as is described in the second property of CNN). The convolution operation is expressed as:

$$y^{j(r)} = ReLU(b^{j(r)} + \sum_i k^{ij(r)} * x^{i(r)}) \tag{8.1}$$

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

Rectifier Linear Unit, also know as ReLU nonlinearity (i.e., $ReLU(x) = max(0, x)$) is used on the obtained feature maps. It is observed that ReLU yields better performance and faster convergence speed when trained by error back propagation [6].

Pooling Layer

As is discussed in the third property of CNN, the pooling layer serves as a mechanism for dimension reduction and feature selection. This layer does not do learning by itself. It takes a small $k \times k$ block from the final feature map of the previous layer and output a single value. The most used pooling methods are max-pooling, where the output is the maximum value of the block, and average pooling, where the output is the average value of the block. There are other pooling methods with good performance on certain tasks [3, 8].

Dropout

Dropout is proposed as an element of the training procedure to reduce overfitting on the training data by preventing coadaptations among neurons [4]. Dropout is performed on each forward passing of a training image, randomly omitting the response

of a neuron from the network with a probability of 0.5. In this way a hidden unit cannot rely on other hidden units being present. It has been shown in [4] that dropout improves the ability of generalization in CNNs on image recognition tasks as well as voice recognition tasks.

8.1.2 Size of the CNN

Size of a typical CNN is usually huge. The winning system of ImageNet 2013 classification contest was a deep convolutional neural network million parameters. The ILSVRC 2012 challenge winning CNN system by Krizhevsky has around 60 million parameters [6]. Overfeat, the ILSVRC 2013 challenge winning CNN, has more than 140 million parameters [12]. Owing to their complexity, these networks are always trained on a GPU machine or GPU clusters for better performance. Are all those parameter needed for image classification? Is there a way to train a compact CNN with the same performance as the state-of-the-art architecture?

8.1.3 Filter suppression and selection

In this subsection, we present a novel way to evaluate the contribution of each filter in a high performance compact Convolutional Neural Network. The filters in the first layer of the proposed CNN are selected from a pretrained larger CNN (2 times larger). The selection is based on ranking the contribution of each filter to the final performance of the network.

Filter Suppression

Filter suppression is used to evaluate the importance of each filter. The term *filter suppression* refers to setting the weight of a specific filter to zero. The performance of the *suppressed* network is then evaluated based on the validation dataset. Contribution of this filter is calculated based on the difference between the error recognition rates before and after filter suppression:

$$Contribution = ERR\_suppressed - ERR\_original \tag{8.2}$$

where ERR stands for error recognition rate, which is the percentage of error recognition in the validation set.

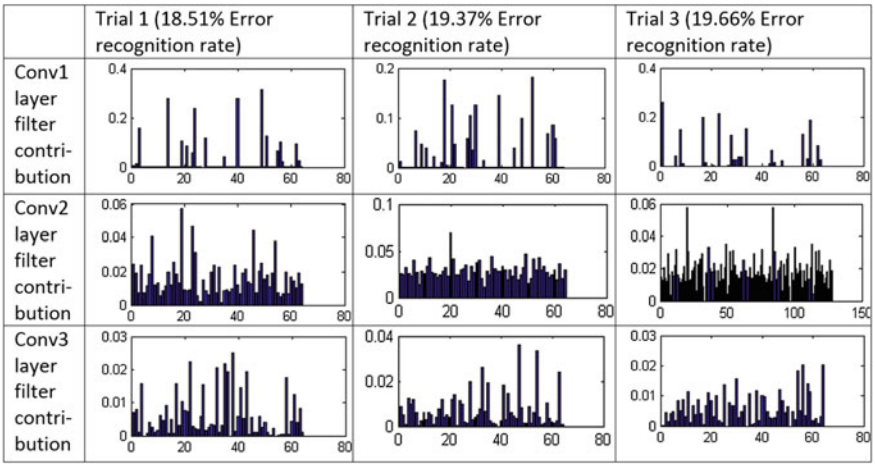

Fig. 8.2 Contribution evaluation for three convolutional neural networks trained on CIFAR-10. For each figure, the x-axis is the index of the filters examined, and the y-axis is the contribution of that filter to the final recognition rate. The contribution of filters in the first convolutional layer varies drastically, indicating that there redundant filters in this layer. Contribution of higher level filters appears to be more uniform compared to the contribution of the filters in the first convolutional layer. The dead filters (more than 50 %) in Conv1 layer can be removed without affecting the final performance

Figure 8.2 shows the contribution evaluation result of three CNNs (with three convolutional layers of the same size) trained on CIFAR-10 dataset. These CNNs are initialized with different parameter (randomly generated) but trained with the same data. The evaluation reveals two important properties of the filters inside a CNN:

1. A large CNN network, though yields good performance during testing, has a considerable amount of *dead* filters in Conv1 layer. By *dead* filters we mean those filters with contribution of 0 % to the recognition rate on the validation dataset. The weight inside those filters can be set to zero without affecting the overall performance of the network.
2. Filters of higher level layers, i.e., Conv2 layer and Conv3 layer, have more averaged contributions to the final performance compared to the filters in the first convolutional layer.

Filter Selection

It is possible that the dead filters in the lower layers, though useless when suppressed individually, are actually important for classification when they are combined together in higher layers. To test that hypothesis, all dead filters are removed in the tested network, including weights that connect the corresponding layer1 feature

Table 8.1 Filter selection result

		Conv1	Conv2	Conv3	Fully Connect	Size	ERR (%)
Original network	Filter size	5 × 5 × 3	5 × 5 × 64	5 × 5 × 64	7 × 7 × 64	240960	18.51
							19.37
							19.66
	Num. of filter	64	64	64	10		
Network without dead filters	Filter size	5 × 5 × 3	5 × 5 × 32	5 × 5 × 64	7 × 7 × 64	187360	18.51
							19.37
							19.66
	Num. of filter	32	64	64	10		

Filter selection result on three randomly initialized networks. The dead filters can be removed without affecting performance of the original network, making the network more compact. ERR stands for error recognition rate

map. The recognition rate, as is shown in Table 8.1, remains unchanged compared to the recognition rate of the original network.

8.2 Compact CNN with Color Descriptor

As is discussed in previous section, CNNs give extraordinary performance on image recognition tasks at the cost of extremely large networks powered by GPUs. The large size of CNNs makes it hard to implement such a system onto a mobile device with limited computational resources. Filter suppression and selection reveals that a CNN by itself is not fully exploiting the lower level information from the input images, generating the dead filters as is shown in Table 8.1. Is there a way to maintain the performance while keeping the network small? In this section we present a compact CNN combined with histogram color descriptor. The proposed solution has a recognition accuracy on par with the state-of-the-art CNNs, while achieving significant model memory footprint reduction. Due to these benefits, the proposed solution is being deployed to the mobile devices.

8.2.1 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 [13]. 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 for device-based image classification task. 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 [10].

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 cut down the size of the basic feature detectors (i.e., layer 1 of the CNN). The proposed architecture is introduced in the following section.

8.2.2 Convolutional Neural Networks

We train two CNNs with different number of filters in the first layer: an original version and a compact version. The 'original' network is the exact replicate of the CNN reported in [5], which gives a final error recognition rate of 13 % using multiview testing on CIFAR-10. In this work, however, we only use single view testing when reporting the final result for both the original CNN and compact CNN.

We use the architecture of Krizhevsky et al. [6] to train the *original* CNN in the experiments. We then modified layer 1 by changing the filter size (from $5 \times 5 \times 3$ to $5 \times 5 \times 1$) and the number of filters (from 64 to 32) in later experiments. The details of the experiments are introduced in the next section.

Both the original and the compact CNNs have four convolutional layers. Table 8.2 shows the details of the two networks when trained on cropped images from the Samsung Mobile Image dataset. Our compact CNN is marked in bold font to show the difference. There are only 32 filters 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 second convolutional layer). The final compact CNN has 40 % less parameters to tune compared to the original version.

8.2.3 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

Table 8.2 Original and compact CNN architecture

	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	$24 \times 24 \times k$	$24 \times 24 \times 64$	$12 \times 12 \times 64$	$12 \times 12 \times 64$	$6 \times 6 \times 64$	$6 \times 6 \times 32$	$6 \times 6 \times 32$	31×1
Compact input size	*$24 \times 24 \times 1$*	*$24 \times 24 \times 32$*	*$12 \times 12 \times 32$*	*$12 \times 12 \times 64$*	*$6 \times 6 \times 64$*	*$6 \times 6 \times 32$*	*$6 \times 6 \times 32$*	*31×1*
Filter size	$5 \times 5 \times k$		$5 \times 5 \times 64$		$3 \times 3 \times 64$	$3 \times 3 \times 32$	$6 \times 6 \times 32$	
Compact filter size	*$5 \times 5 \times 1$*		*$5 \times 5 \times 32$*		*$3 \times 3 \times 64$*	*$3 \times 3 \times 32$*	*$6 \times 6 \times 32$*	
Original filter num	64		64		32	32	31	
Compact filter num	*32*		*64*		*32*	*32*	*31*	
Pool size		3×3		3×3				
Stride	1×1	2×2	1×1	2×2	1×1	1×1		
Output	$24 \times 24 \times 64$	$12 \times 12 \times 64$	$12 \times 12 \times 64$	$6 \times 6 \times 64$	$6 \times 6 \times 32$	$6 \times 6 \times 32$	31×1	

to be important in practice. Minimal improvement on the performance of a support vector machine was observed when switching from RGB color space to HSV color space [1].

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

We experiment with three different configurations of the color histogram:

1. Global histogram, 48 bins.
2. 9-patch histogram, 192 bins. The 9 patches are generated as is shown in Fig. 8.3. 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 are 48, 2 × 24, 2 × 24, and 4 ×24.
3. 9-patch histogram, 384 bins. Numbers of bins are doubled compared to the previous setup.

These experiments on histogram configuration are solely carried out on the CIFAR-10 image dataset. This series of experiment serves as a guideline for our experiment on Samsung Mobile Image Dataset.

8.2.4 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 8.2) with a new fully connected layer and a new softmax

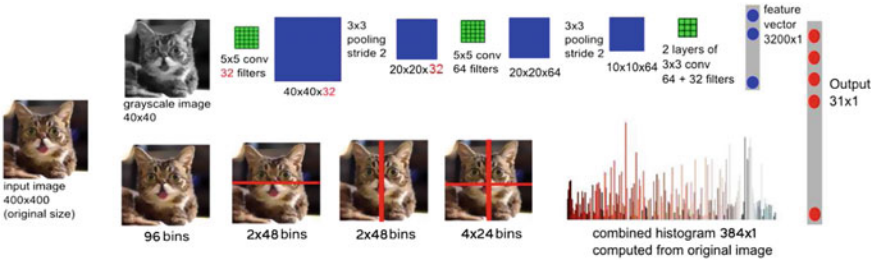

Fig. 8.3 Compact CNN with histogram-based color descriptor. We separate color information from the original image by only feeding the CNN with the grayscale image. Color histogram is combined with the final feature vector. This figure shows how an image from Samsung Mobile Image Dataset is classified as is described in Sect. 8.3.2. 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, selected from the 64 filters in layer 1 of the original network via filter contribution evaluation. The performance of the Compact architecture, therefore, 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

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 number of patches 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 the training 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.

8.3 Experiment

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

1. Cropped images and uncropped images. Training on cropped images (4 corner patches and 1 center patch) means that we feed patches of image into the network instead of the original image. When testing, we feed the network with only the center patch of the 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). This experiment is reported in Sect. 8.3.1.
2. 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 in Sect. 8.3.1. 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. The experiment on this new dataset is reported in Sect. 8.3.2.

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.

8.3.1 Extracting Histogram-Based Color Feature

CIFAR-10 has been heavily tested with many classification methods. Krivzhevsky et al. [6] achieved a 13 % test error rate when using their ILSVRC 2012 winning CNN architecture (without normalization). By generalizing Hinton's dropout [4] into suppression in weight values instead of activation values, Wan et al. [14] reported an error testing rate of 9.32 %, using their modified Convolutional Neural Network DropConnect. Lin et al. [9] replaced the ReLU convolutional layer in Krivzhevsky's architecture [6] with a convolutional multilayer perceptron. They reported a test error rate of 8.8%, currently ranking top on the leader board of classification on CIFAR-10 dataset.

Our experiment in this chapter is still based on Krizhevsky's architecture as is described in [6]. The goal of this paper is to study the contribution of color information to CNN-based image classification, and to seek possible combination between handcrafted 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 [6]. We believe that the combined architecture can also be applied to other CNN variants with few modifications.

Getting Histogram

For device-based image classification, a large histogram vector means heavier load for computation. Therefore we only extract a global histogram of a small amount of 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 the previous section.

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 384, then the number of bins of each patch would be: 96 bins from the entire image, 48×4 bins from the left half, the right half, the top half and the bottom half, 24×4 bins from the upper left corner, the upper right corner, the lower left corner and the lower right corner. This procedure is shown in Fig. 8.3. The intention is to precisely reflect the global color information as well as the local color distribution in the extracted features.

Table 8.3 Different histogram configuration result on uncropped images using original CNN (on CIFAR-10)

Input image and hist config.	Top-1 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

Training Methods

Although our CNN architecture is similar to Krivzhevsky's network, we modify some parts of the training procedures in [6] to suit our needs.

As is shown in Table 8.3, 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 led to significantly improved performance. One important guideline we observed is that a more detailed histogram (384 bins) gives better classification result compared to rough color information.

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.

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 %. However, the original CNN has 146,368 parameters due to the large number of filters in layer 1 and layer 2. The compact CNN trained on grayscale images has less filters in layer 1 and thus 50 % compared to the original CNN, while the error recognition rate rises only by 1 %. As a result, the proposed architecture maintains high performance, while the size of the architecture is 40 % smaller.

8.3.2 Samsung Mobile Image Dataset

The Samsung Mobile Image Dataset is a large scale collection of mobile phone photographs collected at Samsung Research America. There are 31 classes, with a total 82181 images of different sizes and resolutions.

Class names together with sample images of each class are shown in Fig. 8.4. Instead of just training the network to recognize if a person is in the image, the network is also required 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

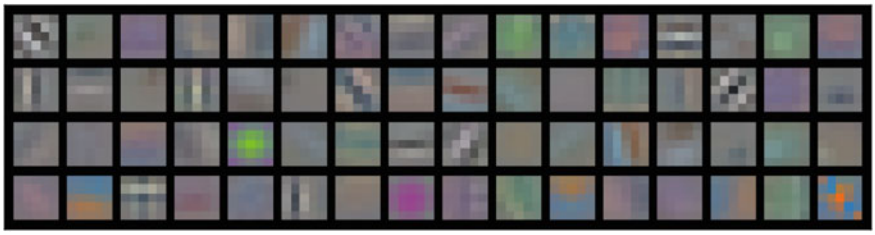

Fig. 8.4 Sample images for Samsung Mobile Image Dataset. This hierarchical image dataset has unclear boundaries among categories. The first level category is presented by colored ovals. Second level categories are presented by the label and a random sample from the training dataset

'food part 2' contains meat and other foods on a plate; the class 'food part 3' consists of pictures about foods on tables. Details of each class can be found in Table 8.6.

We split the dataset by assigning 10 % of the images to the testing set, 10 % to a validation set and 80 % to the training set. After the 384 bins histogram is extracted, each image is then resized 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 8.2. Note that the input image size should be modified accordingly.

Getting Histogram

As the original image contains more detailed information due to the increased image resolution, a global histogram vector is not sufficient to describe the color information with high accuracy.

Guided by the result from our first experiment, we extract a color descriptor of length 384 by concatenating histogram feature vectors from 9 patches of the image as is described in previous experiment (Table 8.4).

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 [6] may lead to confusion

Table 8.4 Cropped image test result (on CIFAR-10)

Architecture (all on cropped images)	Top-1 error rate (%)	Number of parameters
Grayscale (original)	18.10	143168
Grayscale (compact)	18.95	91168
Grayscale (compact) 9 patch hist (384 bins)	*16.55*	*95008*

Table 8.5 Samsung mobile image test result

Architecture (all on cropped images)	Top-1 error rates (%)	Number of parameters
Grayscale (original)	26.08	230848
Grayscale (compact)	26.06	178848
Grayscale (compact) 9 patch hist (384 bins)	*22.80*	*186528*
Dense SIFT aggregation	30.61	–

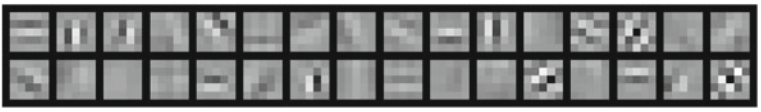

Fig. 8.5 Compact CNN layer 1 filter. There are only 32 filters in layer 1 of the proposed architecture. The network learns basic features as edges and corners from the grayscale input. Network trained on grayscale images from Samsung Mobile Image Dataset

when the network needs to distinguish upper body from whole body (class 9 and 10 in Table 8.6). Therefore we flip the images from the uprightwhole class horizontally at a 0.5 probability. The images are then resized and zero-padded to fit the input size of the network (40×40).

Experiment Result

The error recognition rates of different configurations are reported in Table 8.5.

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 Figs. 8.5 and 8.6.

It can also be seen from the result that color information boosts the performance of the 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 more compact compared to the original version.

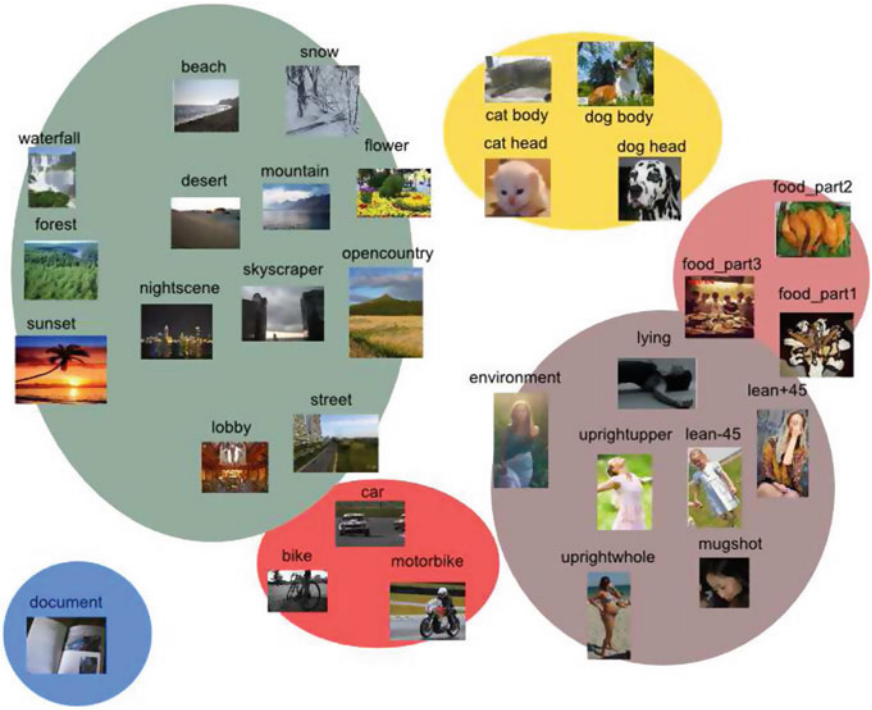

Fig. 8.6 Original CNN trained on RGB images from Samsung Mobile Image Dataset. The network deploys most of its resources in finding color gradient, compared to the filters learned in CNN trained on grayscale images

8.4 Conclusions

In this chapter we introduce the convolutional neural network for image classification. Convolutional neural networks give state-of-the art performance but its application is limited due to its large memory footprint. We present a novel architecture to minimize the size of the network. The proposed architecture combines handcrafted global color information with a convolutional neural network pretrained with thumbnail grayscale images. The proposed architecture has similar recognition capacity compared to state-of-the-art CNNs, quite ahead of the traditional dense SIFT aggregation solution, but with a much smaller network size and complexity that can fit on the mobile devices. We apply our network to Samsung Mobile Image Dataset, a hierarchically organized image dataset. The experiment shows that carefully designed histogram extractor helps to boost the performance of the convolutional neural network. In future work we are investigating a CNN feature map relearning and top-down CNN complexity reduction solution that can further compact the network and improve the accuracy.

Details about the Samsung Mobile Image dataset are included in Table 8.6.

Table 8.6 Class labels and number of images per class

Level 1	Level 2	# of images	Top-1 error rate (%)	Top-2 error rate (%)
Vehicle	Bike	3097	2.64	1.56
	Motorbike	865	6.41	1.79
	Car	2969	21.78	5.37
People	Environment	2713	35.08	6.23
	Lean-45	1271	26.06	10.43
	Lean+45	1277	26.07	10.43
	Lying	1005	23.16	11.58
	Mugshot	3625	16.45	6.45
	Uprightupper	4197	37.01	6.49
	Uprightwhole	3336	37.01	6.49
Food	Food part1	3291	50.00	25.00
	Food part2	2926	20.18	6.02
	Food part3	3168	10.94	3.12
Documents	Document	3080	6.21	3.73
Pets	Cat body	3717	19.13	8.47
	Cat head	3521	5.37	3.95
	Dog body	3769	22.13	10.36
	Dog head	3158	10.39	3.58
Scenery	Flower	3577	4.24	2.12
	Mountain	2838	49.05	13.74
	Skyscraper	2549	49.44	9.20
	Opencountry	1829	31.56	13.78
	Snow	1955	38.34	9.20
	Street	1966	41.82	11.27
	Sunset	2350	60.12	8.90
	Waterfall	1012	4.82	2.19
	Beach	2874	45.26	7.51
	Desert	873	22.22	8.72
	Forest	2667	25.00	5.62
	Lobby	2298	11.48	6.56
	Nightscene	3050	45.51	9.55

References

1. Chapelle, O., Haffner, P., Vapnik, V.N.: Support vector machines for histogram-based image classification. IEEE Trans. Neural Netw. **10**(5), 1055–1064 (1999)
2. Ciresan, D., Meier, U., Schmidhuber, J.: Multi-column deep neural networks for image classification. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, pp. 3642–3649 (2012)

3. Deng, L., Abdel-Hamid, O., Yu, D.: A deep convolutional neural network using heterogeneous pooling for trading acoustic invariance with phonetic confusion. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, pp. 6669–6673 (2013)
4. Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580 (2012)
5. Krizhevsky, A.: Learning multiple layers of features from tiny images. Unpublished
6. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012)
7. LeCun, Y., Denker, J., Henderson, D., Howard, R.E., Hubbard, W., Jackel, L.D.: Handwritten digit recognition with a back-propagation network. In: Advances in Neural Information Processing Systems, Citeseer (1990)
8. Lee, H., Grosse, R., Ranganath, R., Ng, A.Y.: Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In: Proceedings of the 26th Annual International Conference on Machine Learning, ACM, pp. 609–616 (2009)
9. Lin, M., Chen, Q., Yan, S.: Network in network. CoRR, abs/1312.4400, 2013
10. Pass, G., Zabih, R.: Histogram refinement for content-based image retrieval. In: Proceedings of the 3rd IEEE Workshop on Applications of Computer Vision, WACV'96. IEEE, pp. 96–102 (1996)
11. Sermanet, P., Chintala, S., LeCun, Y.: Convolutional neural networks applied to house numbers digit classification. In: 21st International Conference on Pattern Recognition (ICPR). IEEE, pp. 3288–3291 (2012)
12. Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., LeCun, Y.: Overfeat: Integrated recognition, localization and detection using convolutional networks. arXiv preprint arXiv:1312.6229 (2013)
13. Swain, M.J., Ballard, D.H.: Indexing via color histograms. In: Active Perception and Robot Vision. Springer, pp. 261–273. (1992)
14. Wan, L., Zeiler, M., Zhang, S., LeCun, Y., Fergus, R.: Regularization of neural networks using dropconnect. In: Proceedings of the 30th International Conference on Machine Learning (ICML-13), pp. 1058–1066 (2013)
15. Xiao, J., Hays, J., Ehinger, K.A.: Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In: 2010 IEEE conference on Computer vision and pattern recognition (CVPR). IEEE, pp. 3485–3492. (2010)
16. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional neural networks. arXiv preprint arXiv:1311.2901 (2013)