An Adaptive Descriptor Design for Object Recognition in the Wild

Zhenyu Guo, Z. Jane Wang
Dept. of ECE, University of British Columbia
2332 Main Mall
Vancouver, BC Canada V6T 1Z4
{zhenyug, zjanew}@ece.ubc.ca

Abstract

Digital images nowadays show large appearance variabilities on picture styles, in terms of color tone, contrast, vignetting, and etc. These 'picture styles' are directly related to the scene radiance, image pipeline of the camera, and post processing functions (e.g., photography effect filters). Due to the complexity and nonlinearity of these factors, popular gradient-based image descriptors generally are not invariant to different picture styles, which could degrade the performance for object recognition. Given that images shared online or created by individual users are taken with a wide range of devices and may be processed by various post processing functions, to find a robust object recognition system is useful and challenging. In this paper, we investigate the influence of picture styles on object recognition by making a connection between image descriptors and a pixel mapping function g, and accordingly propose an adaptive approach based on a g-incorporated kernel descriptor and multiple kernel learning, without estimating or specifying the image styles used in training and testing. We conduct experiments on the Domain Adaptation data set, the Oxford Flower data set, and several variants of the Flower data set by introducing popular photography effects through post-processing. The results demonstrate that the proposed method consistently yields recognition improvements over standard descriptors in all studied cases.

1. Introduction

Digital images can be different in terms of color tones, contrast, clarity, vignetting, and etc. Here we refer such characteristics of digital images as **picture styles**. With the popularity of photo editing and sharing services such as Instagram, Facebook and Flickr that are available on mobile devices, many digital images generated by users nowadays are captured by a wide range of devices (e.g., smart phones and digital slrs) and processed using different photography effect filters (e.g., "lomo-fi" and "lord-kelvin" available in Instagram) to get distinct picture styles with strong personal artistic expressions. Recall that the goal of object recogni-

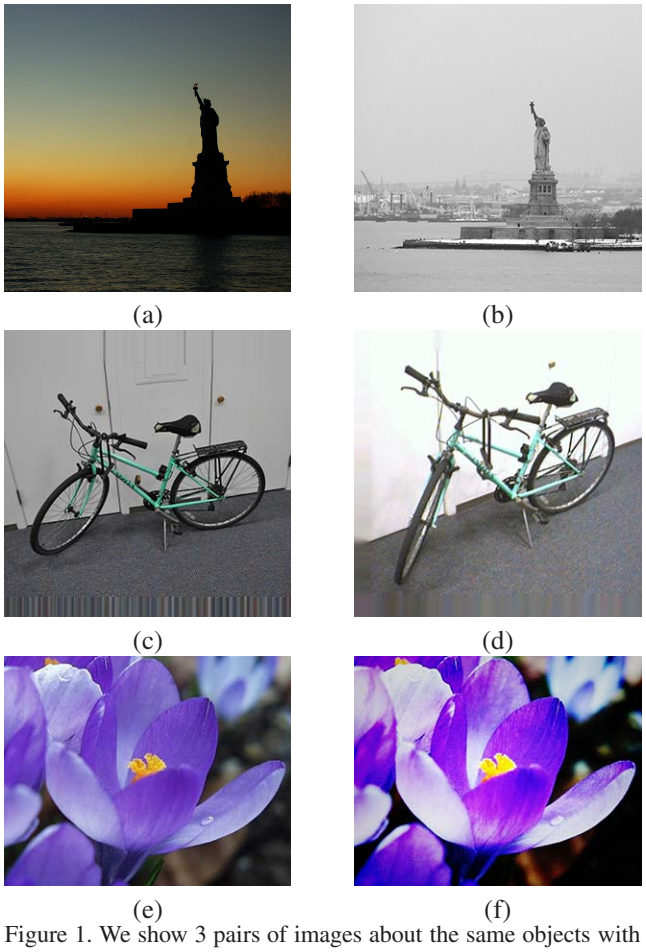

Figure 1. We show 3 pairs of images about the same objects with different picture styles. The differences between (a) and (b) are mainly due to different scene radiances (illumination condition). (c) and (d) are of the same object and taken under the same condition by a digital SLR and a webcam respectively, representing two different image pipelines. (f) is an image obtained by applying Instaram™ *lomo-fi* effect filter as a post-processing step to image (e), representing one specific photography effect.

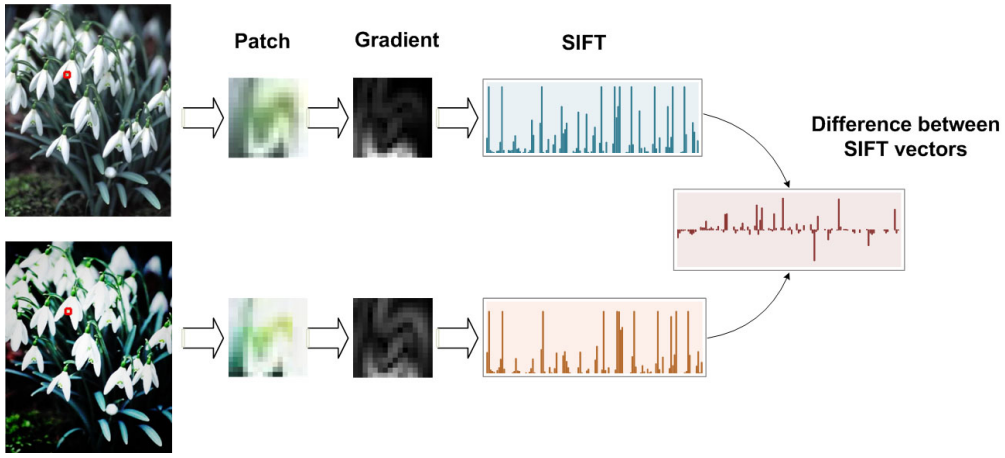

Figure 2. In the upper left is an original image from the Oxford Flower data set. In the lower left is the *lomo-fi* version of the image. We select two regions at the same location from the two images (indicated by red boxes), and show the pixel patch, gradient, and SIFT descriptor for each of them. We then plot the difference between two descriptors in the right.

tion is to recognize natural scenes [14], daily objects [5], or fine-grained species [18, 23] based on digital images, it is natural to extend the scope of object recognition from standard laboratory images to photos in the wild for daily use. Although there are a large number of picture styles, their contributing factors can be separated into 3 major categories: (1) scene radiance, (2) image pipeline, and (3) post processing. In Fig. 1, we show three pairs of images of the same objects to illustrate different picture styles.

To illustrate the connection between image descriptors with picture styles, we take an image from the Oxford Flower data set and process it with a popular *Instagram* effect filter: *lomo-fi*. We select two patches at the same locations for these two images respectively, and compute the gradients and SIFT descriptors of the patches, which are shown in Fig. 2. Although these two image patches are almost the same except the color tones, we note that the resulting SIFT descriptors differ with each other about 33% in terms of l_2 norm, which probably will make them be quantized into two dictionary words in the bag-of-word model. Since the difference is significant for two images that are almost identical in content, it is reasonable to assume that the difference could be more significant for two content-different images with different picture styles within one object class. Therefore, when images used for training and testing don't have similar picture styles, the accuracy of object recognition will degrade. Among the previous related literature, only *Domain Adaptation* (DA) considers the situation [19] where some images are taken by a Digital SLR and the rest are taken by a webcam under similar conditions (e.g. (c) and (d) in Fig. 1), and images used in training and in testing are taken by different devices.

Although the DA touches the picture style issue by considering two sets of images from different devices as two domains, in their algorithms the domain lable of an image has to be specified. However, in real world applications, images collected from Internet have no "domain labels", and the training /testing sets are always mixtures of images with various picture styles. Furthermore, more picture styles can be created by users through post-processing (e.g. Instagram users or iphone camera app users) besides the ones due to different cameras. Therefore, with a more general setting than DA, developing robust object recognition algorithms becomes useful and challenging, which should overcome the difficulties introduced by different picture styles without knowing the style information.

In this paper, we study this general object recognition problem with a focus on picture-style-considered descriptor design. Existing approaches usually ignore the differences of picture styles when computing the standard descriptors, and then try to reduce the influences of picture styles in the corresponding feature spaces. Such indirect methods are limited by the feature spaces and always require the style information of the images (e.g. the domain labels in DA). In this paper, we tackle the problem in a direct way. Suppose a set of images is denoted by A. For an image $I \in A$, we define a pixel mapping function $g : [0, 255] \rightarrow [0, 255]$ that can be applied to all the pixels in I and obtain a new image $g(I)$. Let B denote the set of images such that $\forall I \in A$, $g(I) \in B$. For convenience, we denote $B = g(A)$. From our observation, we assume that the pixel mapping function g would influence the object recognition accuracy when the images used in training and testing are processed by g (which is confirmed later by experimental results in Section 4.2). Therefore, we propose searching an optimal $g*$ that can achieve the best recognition accuracy when all images used are processed by g^*. The searching of g^* could be difficult since there is no clear connections between a general

function g and the empirical risk of the classifier used in object recognition. However, by defining g based on a convex combination of several basis functions, in this paper, we incorporate the pixel mapping function g into image descriptors, and we propose an adaptive descriptor design based on kernel learning. Though we derive the method based on kernel descriptors [2], it is worth mentioning that the proposed approach can be extended to existing standard descriptors as a general framework. In the following, we discuss some related works in Section 1.1. Then we revisit the kernel descriptors in Section 2. We present the proposed method in Section 3 and report the experiments in Section4.

1.1. Related Works

Domain Adaptation is probably the most related area to our problem. In the data set introduced in [19], images from *dslr* and *webcam* are different in picture styles, which is similar to the focus of this paper. Metric learning based methods [13,19], Grassmann manifold based methods [7,8], and output kenrel space based method [10] were proposed. As we stated, these DA methods cannot solve our problem in general situations where the domain label information is unknown and hard to specify for images in the wild. Works in [9,12,26] estimate the model of image pipelines, but such estimations are difficult and have no clear relationships with the descriptor and recognition accuracy. In the area of key point matching, several robust descriptors were proposed, such as DAISY [21], GIH [15] and DaLI [17]. Descriptor learning methods [20, 24, 25] were also developed to determine the parameters of the descriptors through optimization. All these methods are designed for key point matching between image pairs. The different goal leads to descriptors that are not suitable for object recognition, since they are too discriminative to tolerate the within-class variances of object categories.

2. Kernel Descriptor Revisit

The kernel descriptor (KDES) is proposed by Bo et. al. in [2], which gives a unified framework and parametric form for local image descriptors. Let z denote a pixel at coordinate z, $m(z)$ denote the magnitude of image gradient at pixel z, and $\theta(z)$ denote the orientation of image gradient. And $m(z)$ and $\theta(z)$ are normalized by the average values of one patch containing z into $\tilde{m}(z)$ and $\tilde{\theta}(z)$ respectively. The gradient match kernel between two image patches P and Q can be described as

$$k_{grad}(P,Q) = \sum_{z\in P}\sum_{z'\in Q} \tilde{m}(z)\tilde{m}(z')k_o(\tilde{\theta}(z),\tilde{\theta}(z'))k_p(z,z'), \tag{1}$$

where $k_p(z,z') = exp(-\gamma_p||z-z'||^2)$ is a Gaussian position kernel and $k_o(\tilde{\theta}(z),\tilde{\theta}(z')) = exp(-\gamma_o||\tilde{\theta}(z) - \tilde{\theta}(z')||^2$ is a Gaussian kernel over gradient orientations. And $\tilde{m}(z) = m(z)/\sqrt{\sum_{z\in P} m(z)^2 + \epsilon_g}$, where ϵ_g is a small value. Orientation is normalized as $\tilde{\theta}(z) = [sin(\theta(z))cos(\theta(z))]$. To build compact feature vectors from these kernels for efficient computation, [2] presented a sufficient finite-dimensional approximation to obtain finite-dimensioned feature vectors and to reduce the dimension by kernel principal component analysis, which provides a closed form for the descriptor vector $F_{grad}(P)$ of patch P such that $k_{grad}(P,Q) = F_{grad}(P)^T F_{grad}(Q)$. And Bo et. al. [2] also showed that gradient based descriptor like SIFT [16], SURF [1], and HoG [4] are special cases under this kernel view framework.

For the image-level descriptors, Bo and Sminchisescu [3] presented Efficient Match Kernels (EMK) which provide a general kernel view of matching between two images as two sets of local descriptors. And they demonstrated that the Bag-of-Word (BoW) model and Spatial Pyramid Matching are two special cases under this framework. Let X and Y denote the set of local descriptors for images I_x and I_y respectively. $x \in X$ is a descriptor vector computed from patch P_x in image I_x. When applying EMK on top of k_{grad}, we can have the image-level kernel as

$$K_{emk}(I_x,I_y) = \frac{1}{|X||Y|}\sum_{x\in X}\sum_{y\in Y} k_{grad}(P_x,P_y), \tag{2}$$

where $|\cdot|$ is the cardinality of a set. [3] provides a closed-form approximation of the feature vector such that $K_{emk}(I_x,I_y) = \Phi(I_x)^T\Phi(I_y)$, which makes the match kernel can be used in real applications with efficient computation and storage.

3. Proposed Method

3.1. g-incorporated Kernel Descriptor

As stated in Introduction, we want to apply a pixel mapping function g to images used for object recognition. In this section, we will give the relationship between pixels and descriptors under the function g. Since a general function is hard to learn, we define a function $g = \sum_{i=1}^{N} a_i g_i, a_i \geq 0$, which is a convex combination of basis functions. For the convenience of presentation, let's look at an simple example that contains only two basis functions g_1 and g_2, where $g_1(u(z)) = u(z)$ and $g_2(u(z)) = u(z)^2$. Then $g(u(z)) = a_1u(z) + a_2u(z)^2$, where a_1, a_2 are non-negative, z is the position of a pixel from image patch P, and $u(z)$ is the pixel value at z. Let $g(P)$ denote the new patch after applying g on the pixels of P. Now the image gradient at z of $g(P)$ becomes

$$\begin{aligned} \nabla g(u(z)) &= g'|_{u(z)}\nabla u(z) \\ &= (a_1 + 2a_2u(z))\nabla u(z), \end{aligned} \tag{3}$$

where $a_1 + 2a_2u(z)$ is a non-negative real number and $\nabla u(z)$ is a vector. Let $m_g(z)$ and $\theta_g(z)$ be the magnitude

and orientation of $\nabla g(u(z))$, and $m(z)$ and $\theta(z)$ be corresponding values of $\nabla u(z)$. It is clear that $\theta_g(z) = \theta(z)$, therefore $\tilde{\theta}(z) = \tilde{\theta}_g(z)$, which means the orientation is invariant to the pixel mapping function g applied to the image patch. Under the assumption that $a_1, a_2 \geq 0$, we have

$$\begin{aligned} m_g(z) &= ||\nabla g(u(z))|| \\ &= ||(a_1 + 2a_2 u(z))\nabla u(z)|| \\ &= a_1||\nabla u(z)|| + a_2||\nabla u(z)^2|| \\ &= a_1||\nabla g_1(u(z))|| + a_2||\nabla g_2(u(z))||. \end{aligned} \tag{4}$$

Notice that the magnitudes used in k_{grad} are normalized based on local patches, which is important to make the contextual information comparable for different patches. Let $m_1(z)$ and $m_2(z)$ denote $||\nabla g_1(u(z))||$ and $||\nabla g_2(u(z))||$ respectively. To retain the simple convex combination form of $m_g(z)$, we propose a new local normalization

$$\hat{m}_g(z) = a_1 \tilde{m}_1(z) + a_2 \tilde{m}_2(z), \tag{5}$$

where $\hat{m}_g(z)$ denotes the new normalized magnitude of $m_g(z)$, and $\tilde{m}_1(z)$ and $\tilde{m}_2(z)$ are normalized by l_2 norm as mentioned in Section 2. It is clear that $\hat{m}_g(z)$ is also locally normalized and still comparable for different patches. Since the goal of this paper is object recognition, any appropriate local normalization method is acceptable.

Now given two image patches $g(P)$ and $g(Q)$, which are obtained by applying the function g to patches P and Q respectively, we derive the gradient match kernel between them as following

$$\begin{aligned} &\hat{k}_{grad}(g(P), g(Q)) \\ &= \sum_{z\in g(P)} \sum_{z'\in g(Q)} \hat{m}_g(z)\hat{m}_g(z') k_o(\tilde{\theta}_g(z), \tilde{\theta}_g(z')) k_p(z, z') \\ &= \sum_{z\in P} \sum_{z'\in Q} (a_1\tilde{m}_1(z) + a_2\tilde{m}_2(z))(a_1\tilde{m}_1(z') + a_2\tilde{m}_2(z')) \\ &k_o(\tilde{\theta}(z), \tilde{\theta}(z')) k_p(z, z') \\ &= a_1 a_1 \sum_{z\in P} \sum_{z'\in Q} \tilde{m}_1(z)\tilde{m}_1(z') k_o(\tilde{\theta}(z), \tilde{\theta}(z')) k_p(z, z') \\ &+ a_1 a_2 \sum_{z\in P} \sum_{z'\in Q} \tilde{m}_1(z)\tilde{m}_2(z') k_o(\tilde{\theta}(z), \tilde{\theta}(z')) k_p(z, z') \\ &+ a_2 a_1 \sum_{z\in P} \sum_{z'\in Q} \tilde{m}_2(z)\tilde{m}_1(z') k_o(\tilde{\theta}(z), \tilde{\theta}(z')) k_p(z, z') \\ &+ a_2 a_2 \sum_{z\in P} \sum_{z'\in Q} \tilde{m}_2(z)\tilde{m}_1(z') k_o(\tilde{\theta}(z), \tilde{\theta}(z')) k_p(z, z') \\ &= a_1 a_1 k_{grad}(P, Q) + a_1 a_2 k_{grad}(P, Q^2) \\ &+ a_2 a_1 k_{grad}(P^2, Q) + a_2 a_2 k_{grad}(P^2, Q^2), \end{aligned} \tag{6}$$

where P^2 and Q^2 denote the pixel-value-squared patches from P and Q. And it is worth noting that $\hat{k}_{grad}$ above is **different** from the standard k_{grad} in Eq. (1), since we define a different normalization approach in Eq. (5). Since $g = \sum_{i=1}^{2} a_i g_i$, Eq. (6) indicates

$$\hat{k}_{grad}(g(P), g(Q)) = \sum_{i=1}^{2}\sum_{j=1}^{2} a_i a_j k_{grad}(g_i(P), g_j(Q)). \tag{7}$$

Via Eq. (7), we successfully incorporate the pixel mapping function g into image descriptors, which we call g-incorporated KDES.

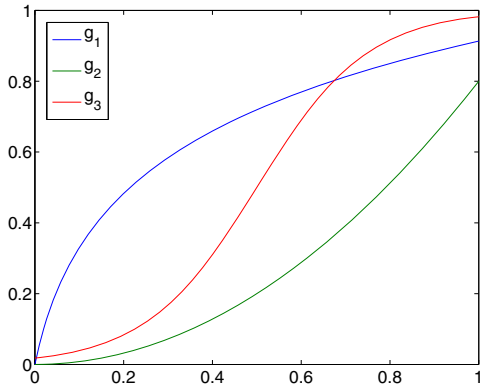

Figure 3. Plots of the proposed basis functions.

To see this connection at the image-level, we plug Eq. (7) into Eq. (2) and have the image-level kernel

$$\begin{aligned} &K_{emk}(g(I_x), g(I_y) = \tfrac{1}{|X||Y|} \sum_{x\in X} \sum_{y\in Y} \hat{k}_{grad}(g(P_x), g(P_y)) \\ &= \tfrac{1}{|X||Y|} \sum_{x\in X} \sum_{y\in Y} \sum_{i=1}^{2} \sum_{j=1}^{2} a_i a_j k_{grad}(g_i(P_x), g_j(P_y)) \\ &= \sum_{i=1}^{2} \sum_{j=1}^{2} a_i a_j (\tfrac{1}{|X||Y|} \sum_{x\in X} \sum_{y\in Y} k_{grad}(g_i(P_x), g_j(P_y))) \\ &= \sum_{i=1}^{2} \sum_{j=1}^{2} a_i a_j K(g_i(I_x), g_j(I_y)) \\ &= \sum_{m=1}^{4} d_m K_m, \end{aligned} \tag{8}$$

where d_m and K_m have one-to-one correspondence to $a_i a_j$ and $K(g_i(I_x), g_j(I_y))$, and the order does not matter since they are exchangeable in the summation. Since we limit a_i to be non-negative when we define g, d_m's are also non-negative. In addition, K_m's are positive definite (PD) kernels, which makes K_{emk} here a convex combination of PD kernels and can be used in standard multiple kernel learning. Therefore, we successfully transfer the problem of searching optimal g^* into the problem of learning the optimal kernel weights through Eq. (8). In general, for N basis functions (i.e., g_i's), there will be N^2 base kernels in Eq. (8).

3.2. Basis Functions g_i's

From Section 3.1, we know that the selection of basis functions (i.e., g_i's) is as important as learning the parameters. By exploring the pixel mapping functions (e.g., photography effect filters) used for photography, we note that Gamma correction and the "S" curve are two major categories of photography effects. Gamma correction can brighten ($\gamma < 1$) or darken ($\gamma > 1$) the images, and the "S" curve can increase the contrast. Although these popular functions are proposed for visual pleasure of photos, we believe that they can also benefit the construction of better image descriptors. For example, brightening can bring back more details in the dark part of an image; Darkening can surpass the irrelevant areas of an object image, since most images are correctly exposed for the center object;

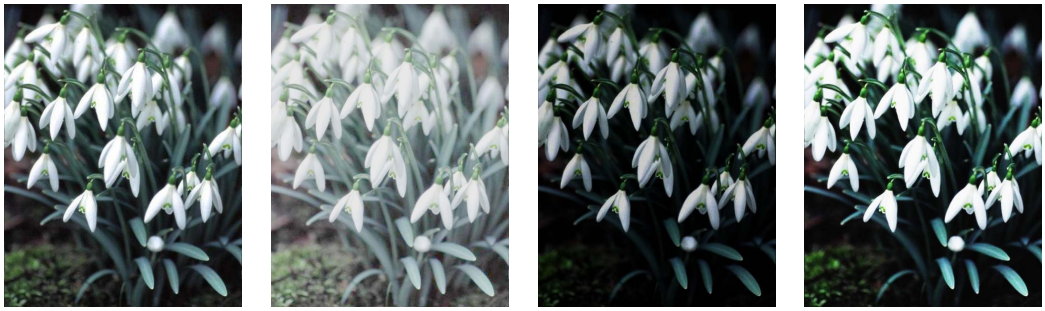

Figure 4. From left to right: the original image I, $g_1(I)$, $g_2(I)$, and $g_3(I)$.

Higher contrast emphasizes the texture and shapes. Therefore, these three types of functions make good candidates for basis functions. However, a power-law function used in Gamma correction contains a free parameter that will be left in the gradient expression and cannot be combined using simple addition, therefore a good approximation is desired. Since the "S" curve doesn't have a standard formulation, we adopt the sigmoid function in our algorithm. In this paper, the three basis functions are:

$$g_1(x) = 0.3 * (\log(2x + 0.1) + |\log(0.1)|) \quad (9)$$

$$g_2(x) = 0.8x^2 \quad (10)$$

$$g_3(x) = \frac{1}{1 + e^{-8(x-0.5)}}, \quad (11)$$

where x takes a value from $[0, 1]$, representing a scaled image pixel value. The plots of these functions are shown in Fig. 3, from which we can see that g_1 could serve as a brightening function, similar to gamma correction when $\gamma < 1$, g_2 has a shape like gamma correction when $\gamma = 2$ and thus could be used as darkening, and g_3 has a "S" shape that increases the contrast by brightening the bright regions and darkening the dark regions. The effects of these functions can be seen from Fig.4.

3.3. Learning of the Parameters

After defining the basis functions, the next question is how to estimate the weight coefficients for object recognition. According to Eq. (8), the image-level kernel can be decomposed as a convex combination of several base kernels. We adopt General Multiple Kernel Learning (GMKL) [22] and put non-negative constraints on the kernel coefficients. We also notice that some weight coefficients can be the same (e.g. $a_1 a_2 = a_2 a_1$), though our experiments show that the results are similar with or without the equal-weight constraint. Since the number of kernels in our algorithm is not large, we use the standard GMKL with $l2$ norm regularization.

3.4. Adaptive Descriptor Design

In this section, we summarize the major steps of the proposed adaptive descriptor design as follows:

Step-1 Process image I from the data set with $\{g_i\}_{i=1}^3$ shown in Eq. (9) (10) (11), to obtain $\{g_i(I)\}_{i=1}^3$.

Step-2 Compute gradient-based descriptors for I and $\{g_i(I)\}_{i=1}^3$ to get 4 descriptors.

Step-3 Build a codebook using K-means by sampling from all training images and all 4 descriptors of each image.

Step-4 Quantize each image from training and testing sets into 4 image-level feature vectors based on the descriptors.

Step-5 For each pair of images, compute linear kernels between any two of their 4 image-level features to get 16 base kernels, as shown in Eq. (8).

Step-6 Train GMKL on 16 base kernels to obtain optimal kernel weights.

Our proposed method does not require prior knowledge on picture styles of training or testing images, and the Adaptive Descriptor Design (ADD) can work as a general framework. For instance, in Step-1, other proper functions can be used here as basis functions, besides the ones we use here. According to the analysis by [2], most gradient-based descriptors, such as SIFT [16], SURF [1] and HoG [4], are special cases of the kernel descriptor, which all can be used in Step-2 to compute descriptors from image patches. In addition, the quantization method used in Step-4 can be chosen from Bag-of-word, Spatial Pyramid Matching and Efficient Match Kernel. In other words, our proposed algorithm can be used widely to improve previous methods which are based on gradient descriptors and SVMs.

We also want to point out that the proposed ADD is essentially a **single** feature method, although multiple kernels are used for estimating the coefficients. For an image I from the data set, it is equivalent to extracting standard descriptors of $g(I)$ and using a single kernel SVM based on these descriptors for classification.

Source	*Target*	standard KDES	ADD_AK	ADD_GMKL
dslr	*webcam*	49.30 ± 1.26	50.28 ± 1.02	**54.81** ± 1.07
webcam	*dslr*	46.67 ± 0.80	48.17 ± 0.99	**50.33** ± 0.79
amazon	*dslr*	47.43 ± 2.79	48.57 ± 2.51	**53.90** ± 2.67

Table 1. Experiments on the DA data set based on the KDES descriptor. The average recognition accuracy in % is reported and the corresponding standard deviation is included.

Source	*Target*	standard SURF	ADD_AK	ADD_GMKL
dslr	*webcam*	37.05 ± 1.72	**41.61** ± 1.05	**42.00** ± 1.16
webcam	*dslr*	30.09 ± 0.81	**36.57** ± 0.75	**36.45** ± 0.49
amazon	*dslr*	34.49 ± 1.30	**40.62** ± 1.59	36.19 ± 2.04

Table 2. Experiments on the DA data set based on the SURF descriptor. The average recognition accuracy in % is reported and the corresponding standard deviation is included.

4. Experiments

In this section, we describe the details of experiments and report the recognition performances of the proposed method when compared with standard gradient-based image descriptors. We conduct object recognition on the Domain Adaptation data set [19] and the Oxford Flower data set [18]. We also process the images from the Oxford Flower data set using several popular photography effect filters in Instagram™[1].

4.1. Domain Adaptation Data Set

The Domain Adaptation data set was introduced by [19], where images for the same categories of objects are from different sources (called domains): *amazon*, *dslr* and *webcam*. As we stated in Introduction, the two domains *dslr* and *webcam* only differ in picture styles which are due to different image pipelines. Applying the proposed ADD algorithm, we adopt KDES + EMK and SURF+BoW two sets of features to demonstrate that ADD can work as a general framework to improve the performances of gradient-based descriptors in general. We follow the experimental protocol used in [13, 19] for semi-supervised domain adaptation. It is worth noting that we don't use any domain-label information to specify the picture styles of images, our proposed method could figure out an optimal descriptor automatically based on the training set.

4.1.1 ADD based on KDES and EMK

We extract KDES descriptors of all the images in three domains and create a 1,500-word codebook by applying K-means clustering on a subset of all 4 types (original + 3 variants for each images) of descriptors from *amazon* domain. And then this codebook is used to quantize 4 types of descriptors of all 3 domains of images using EMK. After obtaining the 16 linear kernels by computing the inner product of every two types of descriptors between two given images, we conduct object recognition experiments using SVMs for: the standard KDES, the average kernel of these 16 kernels (AK), and the GMKL based on 16 kernels. We show the results in Table1, from which we can see that the proposed Adaptive Descriptor Design outperforms the standard KDES in all cases for both the average kernel and an optimal kernel learned by GMKL. Particularly, the ADD_GMKL method improves about 6% from the standard KDES in all cases, which is close to the improvements obtained by domain adaptation methods [7, 11, 13, 19] where domain-label information is used.

4.1.2 ADD based on SURF and BoW

To show the general applicability of the proposed ADD, we follow previous methods [7, 11, 13, 19] to extract standard SURF descriptors from the original and 3 variants of each image, then a 800-word codebook is created from *amazon* domain. All images in 3 domains are quantized by this codebook using Vector-quantization to get Bag-of-Word features. After obtaining 16 linear kernels, we also conduct experiments using the standard KDES, the average kernel, and an optimal kernel learned by GMKL. We report the results in Table 2. The proposed ADD methods also outperform the standard SURF descriptor in all cases. However, in this experiment, the average kernel approach gives better results than that of the GMKL learned kernel in some cases. We think the worse performance of the GMKL based ADD is due to the lack of training, since the SURF descriptors are sparsely extracted from images and only 11 (8 from the source domain and 3 from the target domain) training images per category are used. But the results of ADD_AK and ADD_GMKL are sufficient to show that the proposed Adaptive Descriptor Design can be applied on top of gradient-based descriptors widely, for different tasks.

4.2. Oxford Flower Data Set

Oxford Flower data set [18] contains 1360 images for 17 flower species. To simulate the images used in real world applications, which are taken by different devices and under various pixel-level post-processing, we process the images from the Oxford Flower data set using 3 photography effect filters that are popularly used in Instagram™: *lomo-fi*, *lord-kelvin*, and *Nashville*. Together with the original images, we obtain an image data set of 4 effects. We keep the images generated from the same original images with same IDs. Note that images from the original Oxford Flower data set were collected from many different sources (e.g., they were taken by different devices under different conditions), the factors of scene radiance and image pipelines are already

[1] We use Adobe Photoshop™ action files created by Daniel Box, which can give similar effects as Instagram™.

Figure 5. From left to right: the original image, with *lomo-fi*, *lord-kelvin*, and *Nashville* effects.

taken into consideration. We show an example image and its variants processed by 3 photography effect filters in Fig. 5, and these 4 images have the ID.

Since KDES gives a general framework for gradient-based descriptors, we only use KDES in the experiments in this section. For convenience of expression, we refer the data sets obtained by applying effect filters by picture *styles*. Similar as in Section 4.1.1, we extract KDES for all images from all 4 styles (*original, lomo-fi, lord-kelvin, and Nashville*). We construct a 2,000-word codebook by sampling 4 types of descriptors from the *original* style. Then all images from 4 effects are quantized using EMK with this codebook.

To simulate the real image collections as mixtures of images with different picture styles, we first generate an experimental data set from the 4 styles, then split this data set into training and testing sets. 1) **Experimental data set generation**: M styles are chosen first, from which we want to sample images. For a given ID, only one image is uniformly randomly selected from M styles (i.e. one and only one image from Fig. 5 is sampled) to form a experimental data set. This generation procedure makes sure that images with the IDs will not appear in training and testing together. 2) **Training/testing sets splitting**: after obtaining an experimental data set, we randomly split the set into training and testing sets with equal sizes. Therefore, there are equal numbers of images from each style appearing in training and testing. Different from domain adaptation, images used here in training or testing are not separated according to domain labels, which is more similar to real-world applications where no information of picture styles are available. We perform object recognition using SVMs for the standard KDES, average kernel, and the optimal kernel by GMKL. We report the results for 10 runs of experimental data set for each scenario in Table 3. For each run, an experimental data set is randomly generated and split into training and testing.

From Table 3 we can clearly see that the proposed ADD_GMKL method is superior than the standard KDES in all cases. From the top 4 rows of Table 3, we note that the recognition accuracy decreases when images with different picture styles are used, which confirms the motivation we described in Section 1. In addition, according to the single style experimental results, the pixel mapping function g can influence the recognition accuracy through the computation of the g-incorporated descriptors for all images, which supports the proposed idea that learning an optimal function g^* can improve object recognition when using gradient-based descriptors. Further, when images are uniformly sampled from all 4 styles, the standard KDES descriptor yields worst performance, which is reasonable since the higher diversity in appearances of images leads to larger differences between descriptors of similar image patches of the same objects.

style1	style2	standard KDES	ADD_AK	ADD_GMKL
original	n/a	69.35 ± 2.20	67.76 ± 2.65	**74.32** ± 1.77
original	lomo-fi	65.85 ± 1.66	64.24 ± 1.88	**71.62** ± 1.04
original	lord-kelvin	67.53 ± 1.32	66.06 ± 1.80	**72.82** ± 0.69
original	Nashville	66.44 ± 1.73	64.62 ± 1.39	**71.88** ± 0.72
lomo-fi	n/a	65.12 ± 1.48	63.97 ± 1.62	**69.82** ± 0.70
lord-kelvin	n/a	68.09± 1.38	67.06 ± 1.40	**72.62** ± 1.38
Nashville	n/a	67.03± 1.10	66.85± 1.28	**71.68** ± 1.94
all	n/a	64.56 ± 0.90	63.24 ± 0.58	**69.88** ± 0.51

Table 3. Results on the Oxford Flower data set. The average recognition accuracy in % is reported and the corresponding standard deviation is included.

After demonstrating that picture styles can affect the recognition accuracy, the improved performance of ADD_GMKL shows that our proposed algorithm is an efficient solution. Recall that our ADD can be considered as a single feature method, the proposed ADD_GMKL outperforms the state-of-art [6] by 4% on the original Oxford Flower data set. Therefore, the Adaptive Descriptor Design can be used widely on top of gradient-based descriptors to further improve the recognition accuracy.

We also notice that the ADD_AK method is not better than the standard KDES. We believe it is due to the small size of the codebook (2,000 words). Since 4 types of descriptors are extracted from one image on a dense grid and there are 1360 images in total, this codebook introduces large distortion in quantization, which decreases the performance of the average kernel approach.

5. Conclusion

In this paper, we focus on the effects of different picture styles of images on object recognition. After show-

ing the connection between pixel mapping functions and gradient-based image descriptors, we incorporate the pixel mapping function g into the image descriptor and propose an Adaptive Descriptor Design (ADD) framework for object recognition in the wild. We demonstrate that the proposed g-incorporated ADD can be widely used as a general framework based on popular image descriptors, and the experimental results show the recognition improvements of ADD on the domain adaptation data set, the standard Oxford Flower data set and its variants with different picture styles (photography effect filters).

Acknowledgement

This work was support (in part) by the Canadian Natural Sciences and Engineering Research Council through the NSERC DIVA Strategic Research Network.

References

[1] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (surf). *Computer vision and image understanding*, 110(3):346–359, 2008.

[2] L. Bo, X. Ren, and D. Fox. Kernel descriptors for visual recognition. *Advances in Neural Information Processing Systems (NIPS)*, 7, 2010.

[3] L. Bo and C. Sminchisescu. Efficient match kernel between sets of features for visual recognition. *Advances in neural information processing systems (NIPS)*, 2(3), 2009.

[4] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Conference onComputer Vision and Pattern Recognition (CVPR)*, volume 1, pages 886–893. IEEE, 2005.

[5] L. Fei-Fei, R. Fergus, and P. Perona. One-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4):594–611, 2006.

[6] P. Gehler and S. Nowozin. On feature combination for multiclass object classification. In *IEEE International Conference on Computer Vision (ICCV)*, pages 221–228. IEEE, 2009.

[7] B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2066–2073. IEEE, 2012.

[8] R. Gopalan, R. Li, and R. Chellappa. Domain adaptation for object recognition: An unsupervised approach. In *IEEE International Conference on Computer Vision (ICCV)*, pages 999–1006. IEEE, 2011.

[9] M. D. Grossberg and S. K. Nayar. Modeling the space of camera response functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(10):1272–1282, 2004.

[10] Z. Guo and Z. Wang. Cross-domain object recognition via input-output kernel analysis. *IEEE transactions on image processing*, 22(8):3108–3119, 2013.

[11] I.-H. Jhuo, D. Liu, D. Lee, and S.-F. Chang. Robust visual domain adaptation with low-rank reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2168–2175. IEEE, 2012.

[12] S. Kim, H. Lin, Z. Lu, S. Susstrunk, S. Lin, and M. Brown. A new in-camera imaging model for color computer vision and its application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012.

[13] B. Kulis, K. Saenko, and T. Darrell. What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1785–1792. IEEE, 2011.

[14] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 2169–2178. IEEE, 2006.

[15] H. Ling and D. W. Jacobs. Deformation invariant image matching. In *IEEE International Conference on Computer Vision (ICCV)*, volume 2, pages 1466–1473. IEEE, 2005.

[16] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[17] F. Moreno-Noguer. Deformation and illumination invariant feature point descriptor. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1593–1600. IEEE, 2011.

[18] M.-E. Nilsback and A. Zisserman. A visual vocabulary for flower classification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 1447–1454. IEEE, 2006.

[19] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In *European Conference on Computer Vision (ECCV)*, pages 213–226. Springer, 2010.

[20] K. Simonyan, A. Vedaldi, and A. Zisserman. Descriptor learning using convex optimisation. In *European Conference on Computer Vision ECCV*, pages 243–256. Springer, 2012.

[21] E. Tola, V. Lepetit, and P. Fua. Daisy: An efficient dense descriptor applied to wide-baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(5):815–830, 2010.

[22] M. Varma and B. R. Babu. More generality in efficient multiple kernel learning. In *International Conference on Machine Learning (ICML)*, pages 1065–1072. ACM, 2009.

[23] C. Wah, S. Branson, P. Perona, and S. Belongie. Multiclass recognition and part localization with humans in the loop. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2524–2531. IEEE, 2011.

[24] S. Winder, G. Hua, and M. Brown. Picking the best daisy. In *IEEE Conference onComputer Vision and Pattern Recognition (CVPR)*, pages 178–185. IEEE, 2009.

[25] S. A. Winder and M. Brown. Learning local image descriptors. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[26] Y. Xiong, K. Saenko, T. Darrell, and T. Zickler. From pixels to physics: Probabilistic color de-rendering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 358–365. IEEE, 2012.