What is Happening in a Still Picture?

Piji Li
School of Computer Science & Technology
Shandong University
Jinan, China
Email: peegeelee@gmail.com

Jun Ma
School of Computer Science & Technology
Shandong University
Jinan, China
Email: majun@sdu.edu.cn

***Abstract*—We consider the problem of generating concise sentences to describe still pictures automatically. We treat objects in images (nouns in sentences) as hidden information of actions (verbs). Therefore, the sentence generation problem can be transformed into action detection and scene classification problems. We employ Latent Multiple Kernel Learning (L-MKL) to learn the action detectors from "Exemplarlets", and utilize MKL to learn the scene classifiers. The image features employed include distribution of edges, dense visual words and feature descriptors at different levels of spatial pyramid. For a new image we can detect the action using a sliding-window detector learnt via L-MKL, predict the scene the action happened in and build ⟨action, scene⟩ tuples. Finally, these tuples will be translated into concise sentences according to previously defined grammar template. We show both the classification and sentence generating results on our newly collected dataset of six actions as well as demonstrate improved performance over existing methods.**

***Keywords*-sentence generation; multiple kernel learning; exemplarlets; action classification;**

I. Introduction

Given a picture, psychologists have found that brain react strongly upon seeing human actions. Therefore, for most pictures, humans can prepare a concise description in the form of a sentence relatively easily to identify the most interesting actions. These descriptions are rich because they are in sentence form. However, how to automatically generate sentences to describe what is happening in a still picture?

To tackle the aforementioned problem, we introduce a method to describe what is happening in a still image employing concise sentence as shown in Figure 1. Our contributions include: We define a group of visual discriminative instances for each action class which we called "Exemplarlets" to study this problem. We treat the sentence generation problem as two classification problems: we introduce a Latent Multiple Kernel Learning (L-MKL) method to learn the action detectors from "Exemplarlets" and utilize MKL to learn the scene classifiers respectively. We introduce a simple and effective method to map the ⟨action, scene⟩ tuples to sentences according to previously defined grammar template. We evaluate the sentence generating results on our newly collected dataset of six actions as well as demonstrate improved performance over existing methods.

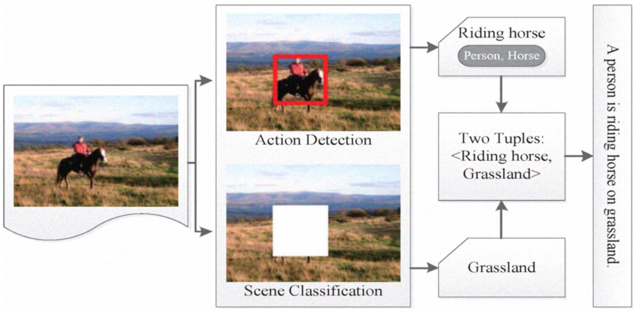


Figure 1. Illustration of our proposed sentence generation framework.

There are few attempts to generate sentences from visual data. [11] present an more sophisticated image parsing to text description (I2T) framework that generates text descriptions of image and video content based on image understanding from a complex database. [5] describe a system that compute a score linking an image to some manually annotated sentence. These methods generate a direct representation of what objects exist and what is happening in a scene, and then decode it into a sentence. However, it is questionable whether the output of any object recognition algorithm is reliable enough to be directly used for event sentence generation.

Action recognition from still images has not been widely studied with the exception of few related papers focused on specific domains, such as sports actions [10] or people playing musical instruments [9]. The proposed methods [10] have mainly relied on the body pose as a cue for action recognition. Inspired by a more general methodology [3], to deal with various types of actions in still images, we avoid explicit reasoning about body poses and investigate more general classification methods.

The rest of this paper is organized as follows. The event sentence generation framework is proposed in Section II. The experimental results and discussions are provided in Section III. Concluding remarks and future work directions are listed in Section IV.

978-1-4577-0121-4/11/$26.00 ©2011 IEEE

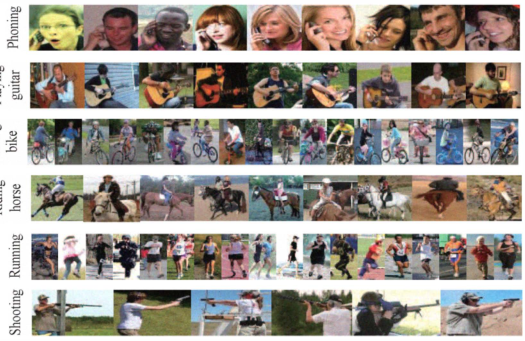


Figure 2. Examples of seed exemplarlets we collected for actions of “phoning” “playing guitar” “riding bike” “riding horse” “running” and “shooting”.

II. Sentence Generation

Figure 1 describes the framework of sentence generation for image understanding. We treat the problem as action detection and scene classification problems and take objects (e.g., person, horse, bike, etc.) as the hidden information of actions (the dark elliptic box).

A. Defining Exemplarlet

An exemplarlet Λ is defined as a sub-image (bounding box) which contains enough visual information for us to identify the action. For instance, the red bounding box in Figure 1 is the exemplarlet for “riding horse”. Let Y be the action label of exemplarlets. In implement, Y is the id set of all the action classes in the database. We denote Λ as $\Lambda = \{A, B, Y\}$, where A is the visual appearance and B describe the size of exemplarlets. B is denoted by $\{b_0, b_1, \cdots b_{K-1}\}$, where K is the number of exemplarlets. The configuration of the k-th exemplarlet b_k is represented as $b_k = (h_k, w_k)$, where (h_k, w_k) is the height and weight value of the k-th exemplarlet.

The exemplarlets we manually selected and segmented from several web image collections (Google[1], Bing[2], Flickr[3]) are called “Seed Exemplarlets”, such as the ones illustrated in Figure 2. In this paper, we just utilize the manually collected exemplarlets to train the action detectors. To name a few, exemplarlets in first row of Figure 2 are the visualization of action query “phoning”.

B. Action Detection

Action detection is the most important component of mapping images to $\langle$action, scene$\rangle$ tuples. For an new input image I, the goal of the detection procedure is to find the “hot region”of I. We zoom I into $|L|$ scales at first, i.e., build the image pyramid. Then we will run the detection algorithm on candidate regions via a sliding-window at the different pyramid level L employ each action detector D. A candidate region R, is a sub-image (window) of the input image I. We can express R as:

$$(I, A, B, P, l, v_c) \tag{1}$$

where I is the parent image of R, A is the visual appearance, B is the size of R (i.e., $height \times width$). P is the position of R on image I and we record the left-top corner (lt_x, lt_y) of the sliding window. $l \in L$ is the pyramid level of image. v_c is the likelihood (score) of R for action class c, and can be computed via Eq. (2) as $D_c(R, Y; \Theta)$, where D_c is the detector learned for each action class c.

$$v_c = \sum_{c \in \mathcal{Y}} D_c(R, Y; \Theta) \cdot \mathbf{1}_c(Y) \tag{2}$$

where $\mathbf{1}_c(Y)$ is an indictor that takes the value 1 if $Y = c$, and 0 otherwise. We take the detectors learning procedure as multi-classification problem, then $\mathcal{Y} = \{1, 2, i, i+1, \cdots, |C|\}$. We will use D represent action detectors in remaining sections. We assume $D(R, Y; \Theta)$ takes the following form:

$$\begin{aligned} D(R, Y; \Theta) &= D(A, B, P, l, Y; \Theta) \\ &= \Theta^T \Psi(A, B, P, l, Y) \end{aligned} \tag{3}$$

where D is parameterized by Θ, $\Psi(A, B, P, l, Y)$ is a feature vector observed from the candidate R. A is the visual appearance of R and we treat B, C and l as latent values.

The hot region R^* of an image I for class c is the region that assigned the maximal v_c. Moreover, we give a constraint condition that the overlap region between the detected hot regions R^* and the real hot region R_0^* is over 50%:

$$\begin{aligned} R^* &= \arg\max_{R \in I}(v_c) \\ \tfrac{\#(R^* \cap R_0^*)}{\#R_0^*} &\geq \tfrac{1}{2} \end{aligned} \tag{4}$$

For the purpose of learning the parameter Θ from “exemplarlets”, given a set of N training examples $\{(\Lambda^{(n)}, Y^{(n)})\}_{n=1}^{N}$, we learn the discriminative and efficient detector function $D : \mathcal{X} \times \mathcal{Y} \to \mathcal{R}$ over an exemplarlet Λ and its class label Y, where $\mathcal{X}$ denote the input space of exemplarlets. During testing and detecting, for an input exemplarlet, we can predict the action type Y^* as:

$$Y^* = \arg\max_{Y \in \mathcal{Y}} D(\Lambda, Y; \Theta) \tag{5}$$

Intuitively, $\cup\Lambda \in \cup R$, i.e., the set of exemplarlets $\cup\Lambda$ is the subset of the set of all candidate regions $\cup R$ in practice. For mathematical representation, we can utilize R instead of Λ and the different is that B, P and l in Λ are fixed constant, e.g., $B = (200, 200)$, $P = (0, 0)$ and $l = 1$.

The detector function $D(\Lambda, Y; \Theta)$ is learnt along with the optimal combination of features and spatial pyramid levels, by using a Latent Multiple Kernel Learning (L-MKL) technique. We just focus on the visual appearance A of

[1] http://images.google.com/
[2] http://images.bing.com/
[3] http://www.flickr.com/

exemplarlets, and let B, P and l as latent values. Therefore, $D(\Lambda, Y; \Theta)$ in Eq. (3) can be rewritten as:

$$\begin{aligned} D(\Lambda, Y; \Theta) &= \sum_{i=1}^{N} \theta_i [K(\Lambda, \Lambda^i), Y_i] \\ &= \sum_{i=1}^{N} \theta_i [K(A, A^i), B, P, l, Y_i] \end{aligned} \tag{6}$$

where A^i, $i = 1, 2, \cdots N$ denote the feature descriptors of N training exemplarlets, and K is a positive definite kernel, obtained as a liner combination of histogram kernels by η:

$$\begin{aligned} &K(A, A^i) = \sum_{k=1}^{\#A} \eta_k k(A_k, A_k^i) \\ &\sum_{k=1}^{\#A} \eta_k = 1 \end{aligned} \tag{7}$$

where $\#A$ is the number of features to describe the appearance of exemplarlets. L-MKL learns both the coefficient θ_i and the histogram combination weights $\eta_k \in [0, 1]$.

Many research work [6], [9] observe that the histogram intersection kernel performs better than the other kernels. Therefore, we employ multiple histogram intersection kernel in this paper:

$$k(A, A^i) = \sum_{i=1}^{N} \min(A, A^i) \tag{8}$$

Since the latent variables B, P and l of exemplarlet Λ are constant, therefore, in the procedure of training, we treat the L-MKL as the classical MKL method [8]. In the procedure of action detection, we can employ the method of iteration for the variable B, P and l. In our experiments, we utilize MK-SVM as the learning method. The detection rule with latent variables as shown in 9

$$f_\Theta(R) = \underset{(Y,H) \in \mathcal{Y} \times \mathcal{H}}{\arg\max} \; D(R, Y; \Theta) \tag{9}$$

where $H = \{B, P, l\}$ and $H \in R$, i.e., H is the set of latent variables of candidate region R.

C. Scene Classification

The other important element of ⟨action, scene⟩ tuple is "scene", which used to describe where the action is happening. For a new image I, e.g., the first image shown in Figure 1, after the action detection procedure, the hot region R^* is detected as the red box in Figure 1. In our work, we treat the difference of I and R^* as the scene (background):

$$S = I - R^* \tag{10}$$

For the purpose of improving the scene classification performance, we draw in to a constraint to make S discriminative enough for classification:

$$\frac{\#S}{\#I} \geq \varsigma \tag{11}$$

where $\#S$ is the area of S and ς is the threshhold. In our experiment we let $\varsigma = 0.4$.

We also employ the Multiple Kernel SVM model in the scene classification task and learn scene classifiers from the dataset we collect.

D. Grammar Template

Interestingly, some actions may contain many hidden information, e.g., the black box shown in Figure 1, "riding horse" contains the hidden objects "person" and "horse", "phoning" contains "person" and "phone". Therefore, when action detection task is addressed, some hidden objects objection tasks would be handled at the same time. It is effectively and efficiently to infer objects from actions in stead of designing objects classifiers. We define a simple sentence grammar template as shown in (12):

$$\begin{aligned} &SENTENCE\langle action, scene\rangle \\ &\to \{\{action\} \pm \{scene\}\} \\ &\to \{\{who + doing + what\} \pm \{where\}\} \end{aligned} \tag{12}$$

where the ± operation in $\pm\{scene\}$ means that if $\frac{\#S}{\#I} < \varsigma$ in Eq. (11), we will omit the scene information for sentence generation. $\{who + doing + what\}$ are all hidden information inferred from $\{action\}$. For convenience, we employ WordNet synsets to compute some word collocation. Finally, we get a concise sentence according to this simple grammar template. For instance shown in Figure 1, the concise sentence is "A person is riding horse on grassland.".

III. Experiments and Results

A. Dataset

We collect about 2400 images in total for six action queries: phoning, playing guitar, riding bike, riding horse, running and shooting. Most of the images are collected from Google Image, Bing and Flickr, and others are from PASCAL VOC 2010 [4] and PPMI dataset [9]. Each action class contains about 400 images. We manually select and segment 100 exemplarlets for each action query and after refined via cross-validation we select top 60 as the seed exemplarlets. The size ($height \times width$) of exemplarlet for each class are 200×200, 200×200, 300×150, 200×200, 300×150 and 200×300. Some examples of exemplarlets are listed in Figure 2.

We also collect about 3000 images in total for fifteen scene categories [6], we add some other scene categories and omit several categories for our purpose, e.g., "grassland" and "sea side".

B. Appearance Descriptors

The descriptors of the appearance of the images are constructed from a number of different feature channels.

Dense SIFT words [6]. Rotationally invariant SIFT descriptors are extracted on a regular 16×16 grid each eight

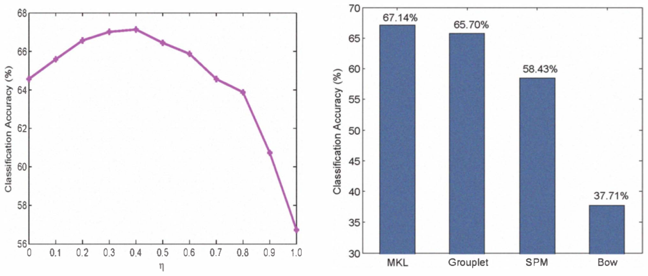


Figure 3. 7-class classification using the normalized PPMI+ images. left: the weight η for the kernel based on the SIFT feature channel, we choose $\eta = 0.4$. right: Classification results of different methods.

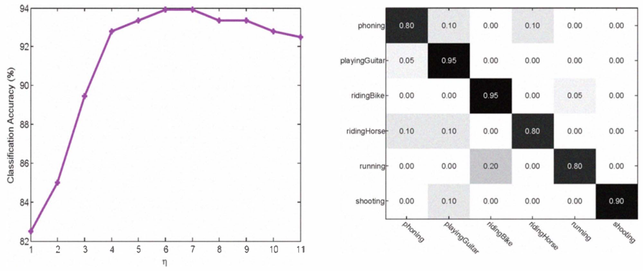


Figure 4. 6-class classification using exemplarlets. left: the weight η for the kernel based on the SIFT feature channel, we choose $\eta = 0.7$. right: Confusion matrix obtained by MKL.

pixels, zeroing the low contrast ones. Descriptors are then quantized in 300 visual words.

Histogram of oriented edges [2]. The Canny edge detector is used to compute an edge map and the underlying image gradient is used to assign an orientation and a weight to each edge pixel p. The orientation angle is then quantized in eight bins with soft linear assignment and an histogram is computed.

Gist [7]: We encode global information of images using gist.

Spatial pyramid. For each feature channel a three-level pyramid of spatial histograms is computed, similar to [6].

C. L-MKL for Action Detection

Multiple kernel learning is the basic learning model in our framework, we must make sure that the detectors learnt via MKL are effective and discriminative. We utilize SVM as the learning method and employ the Matlab package libsvm [1] as the implement of SVM. We compare the MK-SVM method with the-state-of-art [6], [9].

We show results on the datasets Yao and Fei Fei [9] in Figure 3(right). Both BoW and SPM [6] use the histogram representation, where BoW does not consider spatial information in image features while SPM accounts for some level of coarse spatial information by building histograms in different regions of the image. The BoW representation is followed by an SVM classifier with the histogram intersection kernel. As shown in Figure 3(right), the MK-SVM method outperform the approach of [9] and [6] by 1.44% to 8.71%.

For analyzing the discrimination of exemplarlets, we treat all exemplarlets as a six-classification problem. We compute mean accuracy of five times five-fold cross validation via L-MKL-SVM, the classification accuracy is shown in Figure 4. All the accuracy are greater than 80% and the mean accuracy is 86.67%. We believe that the exemplarlets for each action class are discriminative enough to train effective detectors. On a PC with two 2.93GHz CPU, the train task can finish in 5.09 seconds (We neglect the cost for computing visual features).

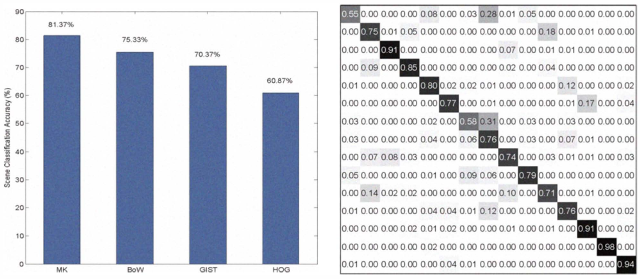


Figure 5. 15-class scene classification. left: compare the multiple kernel method with some other state-of-art features. right: confusion matrix obtained by MK-SVM. The classification accuracy is 81.37%.

D. MKL for Scene Classification

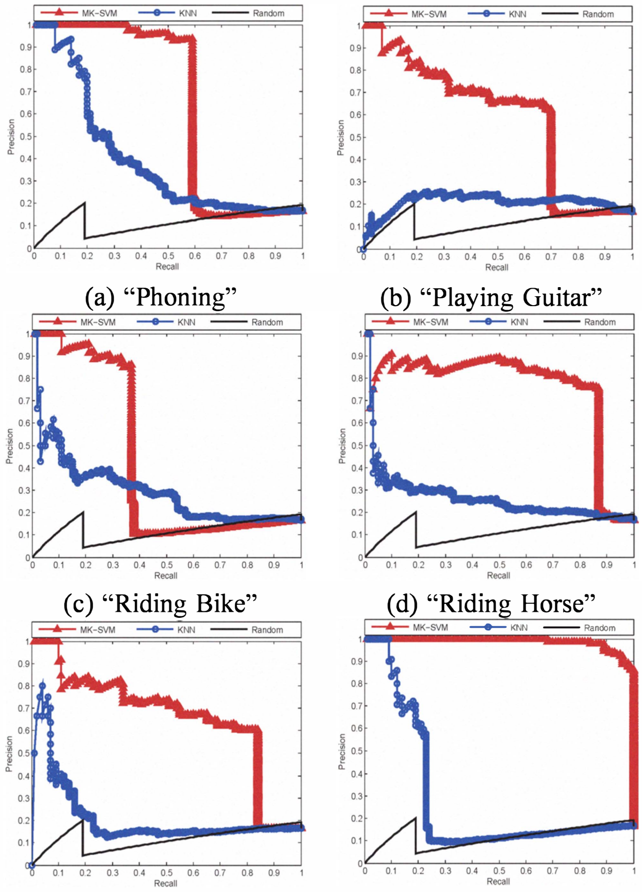
In this section, we report results on fifteen scene categories we collected. We perform all processing in grayscale, even when color images are available. All experiments are repeated ten times with different randomly selected training and test images and the final result is reported as the mean and standard deviation of the results from the individual runs. Multi-class classification is done with the MK-SVM.

The scene classification results are shown in Figure 5. We utilize spatial pyramid SIFT and GIST in the MK-SVM, and compare the performance with some other visual features. It shows that the MK-SVM can improve the scene classification performance.

E. Evaluation of Sentence Generation

All the sentences generated to describe images are simple, therefore, we use the joint classification performance to evaluation sentences. For an new image I, after the action detection and scene classification tasks, we get a $\langle$action, scene$\rangle$ tuple to describe it. We defined here iff both action label and scene label are correct, the concise sentence generated from this $\langle$action, scene$\rangle$ tuple is correct.

$$
\begin{aligned}
&Sentence(I)\\
&=\begin{cases} 1, & if Action(I)=1\ and\ Scene(I)=0\\ & \quad or Action(I)=1\ and\ Scene(I)=1\\ -1, & otherwise \end{cases}
\end{aligned}
\tag{13}
$$



(a) "Phoning" (b) "Playing Guitar"

(c) "Riding Bike" (d) "Riding Horse"

(e) "Running" (f) "Shooting"

Figure 6. The precision-recall cures we used to show the sentence generation performance.

Where $Sentemce(I) = 1$ means that the generated sentence for image I is correct. $Scene(I) = 0$ means that $\frac{\#S}{\#I} < \varsigma$ in Eq. (11), so there is no background for the event happening in image I.

We test our framework on 600 images, and get the precision-recall curve to display the performance, comparing with a KNN method. The results are shown in Figure 6. The precision-recall curves are listed according to the event happening in images, i.e., the six action classes we collect for our experiments: phoning, playing guitar, riding bike, riding horse, running, shooting. It is evident that the proposed L-MKL method is outperform some other methods and is useful in practice.

IV. Conclusion

In this paper we introduces a framework to generate concise sentences to describe still pictures automatically. We employ Latent Multiple Kernel Learning (L-MKL) to learn the action classifiers from "Exemplarlets", and utilize MKL to learn the scene classifiers. We treat some objects as the hidden information of actions and omit the object recognition progress. This methodology could avoid some complex object recognition problems, however, it may lose some of the detail information.

In the future work, we will study models to generate more complex sentences to describe still images and these new sentence will contain complete and accurate information.

Acknowledgments

This work is supported by the Natural Science Foundation of China (60970047,61103151,61173068).

References

[1] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[2] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, volume 1, pages 886–893. IEEE, 2005.

[3] V. Delaitre, L. I., and S. J. Recognizing human actions in still images: a study of bag-of-features and part-based representations. In *BMVC*, 2009.

[4] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88(2):303–338, June 2010.

[5] A. Farhadi, M. Hejrati, M. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, and D. Forsyth. Every Picture Tells a Story: Generating Sentences from Images. *ECCV*, pages 15–29, 2010.

[6] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, volume 2, pages 2169–2178. IEEE, 2006.

[7] A. Oliva and A. Torralba. Building the gist of a scene: The role of global image features in recognition. *Progress in brain research*, 155:23–36, 2006.

[8] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman. Multiple kernels for object detection. In *ICCV*, pages 606–613. IEEE, 2010.

[9] B. Yao and L. Fei-Fei. Grouplet: a structured image representation for recognizing human and object interactions. In *CVPR*, pages 9–16. IEEE, 2010.

[10] B. Yao, A. Khosla, and L. Fei-Fei. Classifying actions and measuring action similarity by modeling the mutual context of objects and human poses. In *ICML*, Bellevue, USA, June 2011.

[11] B. Yao, X. Yang, L. Lin, M. Lee, and S. Zhu. I2T: Image parsing to text description. *Proceedings of the IEEE*, 98(8):1485–1508, 2010.