Annotating Videos From the Web Images

Han Wang, Xinxiao Wu, Yunde Jia
Beijing Laboratory of Intelligent Information Technology, School of Computer Science
Beijing Institute of Technology, Beijing 100081, P.R. China
{*wanghan,wuxinxiao,jiayunde*}*@bit.edu.cn*

Abstract

In this paper, we propose a generic framework for annotating videos based on the web images. To greatly reduce expensive human annotation on tremendous quantity of videos, it is necessary to transfer the knowledge learned from the web images with a rich source of information to video. A discriminative structural model is proposed to transfer knowledge from web images (auxiliary domain) to videos (target domain) by jointly modeling the interaction between video labels and web image attributes. The advantage of our framework is that it allows us to infer video labels using the information from different domains, i.e. the video itself and image attributes. Experimental evidence on UCF Sports Action Dataset demonstrates that it is effective and efficient to use knowledge gained from web images for video annotation.

1. Introduction

With the growing number of videos recorded every second across the world, video annotation becomes a challenge but important computer vision task to understand digital multimedia contents for browsing, searching, and navigation. To deal with the automatic annotation, conventional methods [9][11] usually build a classifier to detect the presence of the concepts in a video clip based on a large corpus of labeled example videos, in which the concept labels are generally obtained through human annotation. However, labeling a video is time consuming and labor intensive. Consequently, classifiers are learned from a limited number of labeled training samples, which are usually not robust and do not generalize well.

Due to the fact that finding enough labeled video data that covers a diverse set of classes is quite challenging, we consider to obtain video knowledge from the web images. The main observation behind our approach is that web images contain a rich source of information, with many activities taken under various conditions. Moreover, it is much more easier to obtain knowledge from a single image than from a video with a mount of frames. In this work, we argue that some videos can be identified by a single monocular image and the support of motion information provided by videos. Therefore, how to accurately annotate videos by making the maximum use of the image data becomes a critical problem.

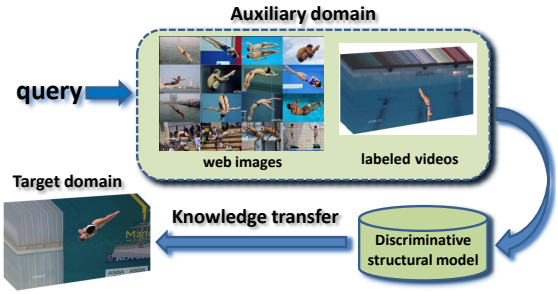

Figure 1. Illustration of framework.

In this paper, we propose a new framework for annotating videos by leveraging a large mount of labeled images and a small set of labeled videos. A large set of images collected from web and a small set of videos together make up the auxiliary domain. We treat the image attribute as auxiliary label with continuous value and propose to use a discriminative structural model to jointly capture the correlation between attributes and video labels, as well as the relationship among different attributes. Consequently, in this model auxiliary label and target label are integrated and learned jointly in a unified framework. Finally, we use this discriminative model learned from auxiliary domain to predict the label of videos from target domain. Fig.1 illustrates our framework.

2. Related work

To the best of our knowledge, not much work has been reported on annotating videos using still images. Konard, et al. [10] proposed to use key poses extracted

from single video frames for action recognition. Their method requires a large amount of training videos, especially for recognizing actions from real world videos. In[6], action models were first learned from loosely labeled web images. This work cannot distinguish actions like "standing-up" and "sitting-down" because it does not utilize temporal information of actions in the image-based model.

For annotating videos in different domains, there is a vast mount of work in the literature. Some approaches [1][2] require explicitly extracted silhouettes from videos in static or uniform backgrounds. Some approaches [5][7][14] work with more realistic videos in the presence of background clutter. Our work does not require much clean video data for training, and it is intended to build a bridge between Internet vision and video annotation.

3. Feature extraction

In this paper, we are given the initial results of a key word query to an image search engine and a small amount of loosely labeled videos. In this work, a training sample is denoted as a tuple $(\mathbf{x}, \mathbf{h}, y)$, where y is the class label of the video. The sample $\mathbf{x} = (x_{img}, x_{vid})$ with x_{img} and x_{vid} representing image part and video part of the sample, respectively. The attribute feature of $\mathbf{x}$ is denoted by a K-dimensional vector $\mathbf{h} = [h_1, h_2, ..., h_K]$, where $h_k \in \mathcal{H}$ indicates the weight of the k-th attribute. For x_{img}, rotation invariant SIFT features [8] are used to better capture the visual similarity within each event type. For x_{vid}, the spatio-temporal interest points [4] are extracted to describe the motion information of video. For both x_{img} and x_{vid}, we employ the bag-of-words approach to generate the final feature vector.

We propose to discover the data-driven attributes by clustering low-level image features $\mathbf{F} = \{\mathbf{f}_1, \mathbf{f}_2, ..., \mathbf{f}_M\}$ with $\mathbf{f}_m \in \mathbf{R^D}$, where M represents the number of images in auxiliary domain and $\mathbf{D}$ is the dimension of image feature vector. N cluster centers are denoted by $C = \{c_1, c_2, ..., c_N\}$, where $c_n \in \mathbf{R^D}$. The weight of j-th attribute h_j for $\mathbf{f}_m$ calculated by $h_i = exp(-d_i)$, where $d_i = \sqrt{|\mathbf{f}_m - c_i|^2}$ stands for the distance between image features and ~~n~~-th cluster center. Consequently, the data-driven attributes ~~we gained~~ are continuous values instead of discrete values.

4. Knowledge transfer

Following the prior terminology, we refer to the web images and labeled videos as auxiliary domain D^A. Specifically, $D^A = D_m^A \bigcup D_v^A$, where D_m^A and D_v^A represent the labeled image data collected from web and labeled video data, respectively. Meanwhile, testing videos are referred as target domain D^T. Given a set of N training samples$\{(\mathbf{x}^{(n)}, \mathbf{h}^{(n)}, y^{(n)})\}_{n=1}^N$, where $\mathbf{x^{(n)}} = (x_{img}^{(n)}, x_{vid}^{(n)})$ with $x_{img}^{(n)} \in D_m^A$ and $x_{vid}^{(n)} \in D_v^A$, our goal is to learn a model to assign the class label y to an unseen test video $\mathbf{x} \in D^T$.

4.1. Model formulation

In this setting, our first attempt is to learn a discriminative structure function $f_{\mathbf{w}} : X \times H \times Y \rightarrow \mathbb{R}$ to annotate videos, where $\mathbf{w}$ is the parameter vector ~~providing a weight for each feature.~~ In testing~~,~~$f_{\mathbf{w}}$ is used to predict the label of a new video $\mathbf{x}$, namely $y^* = argmax_{y^* \in Y} f_{\mathbf{w}}(\mathbf{x}, \mathbf{h}, y)$. We assume that $f_{\mathbf{w}}(\mathbf{x}, \mathbf{h}, y)$ takes the following form:$f_{\mathbf{w}}(\mathbf{x}, \mathbf{h}, y) =$ ~~($~~$\mathbf{w}^T \Phi(\mathbf{x}, \mathbf{h}, y)$~~)~~, where $\Phi(\mathbf{x}, \mathbf{h}, y)$ is the feature vector depending on the sample $\mathbf{x}$, its attribute $\mathbf{h}$ and its class label y. $\mathbf{w}^T \Phi(\mathbf{x}, \mathbf{h}, y)$ is defined as:

$$\begin{aligned}\mathbf{w}^T \Phi(\mathbf{x}, \mathbf{h}, y) = \alpha_y^T \phi(x_{img}) + \sum_{j \in V} \beta_j^T \varphi_j(x_{img}) \\ + \gamma_y^T \upsilon(x_{vid}) + \sum_{j \in V} \theta_{j,y}^T \omega_j(x_{img}) \\ + \sum_{j,k \in \varepsilon} \mu_{j,k}^T \psi(h_j, h_k) + \sum_{j \in V} \eta_{j,y} h_j,\end{aligned} \tag{1}$$

where $\mathbf{w} = [\alpha_y; \beta_j; \gamma_y; \theta_{j,y}; \mu_{j,k}; \eta_{j,y}]$. The details of each term in Eq.(1) are described in the following.

Video class model on static feature $\alpha_y^T \phi(x_{img})$: This term provides the score measuring how well the raw still image feature of video $\mathbf{x}$ matches the event class without considering attributes. $\phi(x_{img})$ represents the image feature vector of the sample.

Global attribute model $\beta_j^T \varphi_j(x_{img})$: This term provides the score of an individual attribute, and is used to indicate the relatively presence of an attribute in the image without consider its event class or other attributes. β_j is a template for predicting the j-th attribute and $\varphi_j(x_{img})$ is derived from $\phi(x_{img}) \cdot h_j$, where $\phi(x_{img})$ is the image feature vector and h_j represents the weight of the j-th attribute.

Video class model on motion feature $\gamma_y^T \upsilon(x_{vid})$: This term measures how well the raw motion feature of video $\mathbf{x}$ matches the event class without considering any attributes. Similarly, $\upsilon(x_{vid})$ is the spatio-temporal feature ~~of the motion part of input samples~~.

Event-specific attribute model $\theta_{j,y}^T \omega(x_{img})$: This term provides the score of the j-th individual attribute given the video class label y. $\theta_{j,y}$ represents a template

for predicting the j-th attribute with special video class label y. Similarly, $\omega(x_{img})$ is defined as $\omega(x_{img}) = \phi(x_{img}) \cdot h_j$.

Attribute interaction model $\mu_{j,k}^T \psi(h_j, h_k)$: In our work, we believe that there are certain dependencies between different attributes. This edge potential $\psi(h_j, h_k) = h_j \cdot h_k$ captures the relationship of the j-th attribute and the k-th attribute.

Event-attribute interaction model $\eta_{j,y} \cdot h_j$: This is a scalar indicates how likely the video class being y and the j-th attribute weight being h_j.

4.2. Learning ~~objective~~

The parameter vector $\mathbf{w}$ learned from the auxiliary domain can be solved by the following objective function:

$$\min_{\mathbf{w},\xi} \lambda \|\mathbf{w}\|^2 + \sum_{n=1}^{N} \xi^{(n)}$$
$$\mathbf{s.t.} \mathbf{w}^T \Phi(\mathbf{x}^{(n)}, \mathbf{h}^{(n)}, y^{(n)}) - \max_{\mathbf{h}} \mathbf{w}^T \Phi(\mathbf{x}^{(n)}, \mathbf{h}, y)$$
$$\geq \Delta(y, y^{(n)}) - \xi^{(n)}, \forall n, \forall y, \quad (2)$$

where λ is the trade-off parameter controlling the amount of regularization, ~~and~~ $\xi^{(n)}$ is the slack variable for n-th training example to handle the case of soft margin, $\Delta(y, y^{(n)})$ is a loss function indicating the cost of misclassifying $y^{(n)}$. Here we typically use the 0-1 loss $\Delta_{0/1}(y, y^{(n)}) = 1$, if $y \neq y^{(n)}$, and otherwise $\Delta_{0/1}(y, y^{(n)}) = 0$.

We rewrite the constrained optimization problem in Eq.(2) as an equivalent unconstrained problem:

$$\min_{\mathbf{w}} L(\mathbf{w}) = \lambda \parallel \mathbf{w} \parallel^2 + \sum_{n=1}^{N} R^n(\mathbf{w}), \quad (3)$$

where $R^n(\mathbf{w}) = \max_y(\max_{\mathbf{h}} \mathbf{w}^T \Phi(\mathbf{x}^{(n)}, \mathbf{h}, y) + \Delta(y, y^{(n)})) - \mathbf{w}^T \Phi(\mathbf{x}^{(n)}, \mathbf{h}^{(n)}, y^{(n)})$. We adopt a convex cutting plane method proposed in [3] to solve the optimization problem in Eq.(3). In order to solve the inference problem $\max_y(\max_{\mathbf{h}} \mathbf{w}^T \Phi(\mathbf{x}^{(n)}, \mathbf{h}, y) + \Delta(y, y^{(n)}))$, we enumerate all the possible video class labels to find the optimal y and infer $\mathbf{h}$ by quadratic optimization given the fixed video class y.

5. Experiments

5.1. Dataset

We evaluate our method using the UCF Sports Dataset [12]. This dataset contains ten different types of sports and consists of 149 real videos with a large intra-class variability. Each sport class is performed in different number of ways, and the frequencies of various sports also differ considerably. We apply nine classes of more complex sports in our experiments: swinging, diving, kicking, weight-lifting, horse-riding, running, skateboarding, swing at high bar(HSwing), golf swinging(GSwing).

To begin, we utilize several query words related to each sports class (i.e. "weight lifting", "horse riding") on web search engines like Google and Yahoo! Image Search, and collect 100 images for each class of the sports as the training image part of auxiliary domain. Fig.2 shows some examples of the web image set.

5.2. Experimental setting

For the static image feature, we build a codebook of 1000 visual words by applying K-means on the SIFT features extracted from image training set. Meanwhile, for the motion video feature, we employ spatio-temporal interest points as the motion features fed into our system. We use the same implementation as in [13]. 1000 interest points are extracted from each video with the spatial and temporal scale parameters σ and τ are empirically set by $\sigma = 2.5$ and $\tau = 2$, respectively. And the size of cuboid around each point is empirically fixed as $7 \times 7 \times 5$. For simplicity, we flatten each normalized cuboid and extract the gray-level pixel values from each normalized cuboid. Principle Component Analysis (PCA) is used to reduce the dimension of appearance feature vector by preserving 98% energy. At the training stage, we select two videos in each sport class as training videos of auxiliary domain to provide training motion information and leave the rest videos for testing. And during the testing stage, we randomly select one frame per video as its key frame representing its static image feature part.

5.3. Results

We conduct experiments to verify the proposed knowledge transfer method. Besides our approach, for labeling sports in videos, we try two different baselines: one-vs-all SVMs and multi-class SVMs with linear kernels. The annotating precision is measured using the mean average precision (mAP) which is the mean precision of all the testing videos. Table 1 shows SVMs results using only video information provided by the small set of videos in auxiliary domain without any knowledge learned from web iamge. As shown in Table 1, our method outperforms both one-vs-all SVMs and multi-class SVMs. This is because our method obtains knowledge from the web images which contain a rich source of information.

Table 2 demonstrates the annotation performance of using the attribute feature and excluding attribute feature to further evaluate the effectiveness of the attribute knowledge learned from web images. We also compare

Figure 2. Example images collected from web.

Table 1. Comparison of our method to other baseline methods.

Method	mAP(%)
one-vs-all SVM	58.9
multi-class SVM	56.7
our method	65.2

Table 2. Annotation precision on motion and attributes.

Method	mAP(%)
transfer without motion	53.7
transfer without attribute	49.8
our method	65.2

the performance without motion feature extracted from videos to that using motion feature. From Table 2, it is interesting to notice that the combination of the attribute feature learned from web images and the motion information extracted from videos significantly improves the recognition performance. Fig.3 illustrates the mAP annotation accuracy of each class.

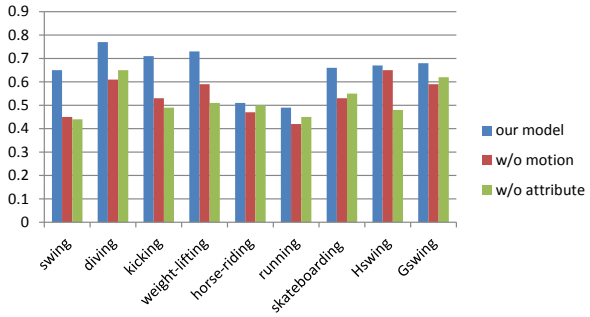

Figure 3. The performance comparison on each class of sports

6. Conclusion

We have addressed the problem of annotating video clips with the help of images collected from web. Our aim is not to compete with action recognition algorithms that work purely on videos, but show-with experimental evidence-that the knowledge obtained from web images can be transferred to annotate the videos. Future work includes finding more referenced images from the web and using less video to accelerate the training process.

References

[1] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. In *ICCV*, volume 2, pages 1395–1402. IEEE, 2005.

[2] L. Cao, Z. Liu, and T. Huang. Cross-dataset action detection. In *CVPR*, pages 1998–2005. IEEE, 2010.

[3] T. Do and T. Artières. Large margin training for hidden markov models with partially observed states. In *ICML*, pages 265–272. ACM, 2009.

[4] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *VS PETS*, pages 65–72. IEEE, 2005.

[5] L. Duan, D. Xu, I. Tsang, and J. Luo. Visual event recognition in videos by learning from web data. In *CVPR*, pages 1959–1966. IEEE, 2010.

[6] N. Ikizler-Cinbis, R. Cinbis, and S. Sclaroff. Learning actions from the web. In *CVPR*, pages 995–1002. IEEE, 2009.

[7] J. Liu, J. Luo, and M. Shah. Recognizing realistic actions from videos in the wild. In *CVPR*, pages 1996–2003. IEEE, 2009.

[8] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[9] M. Naphade. Statistical techniques in video data management. In *Multimedia Signal Processing*, pages 210–215. IEEE, 2002.

[10] K. Schindler and L. Van Gool. Action snippets: How many frames does human action recognition require? In *CVPR*, pages 1–8. IEEE, 2008.

[11] C. Snoek, M. Worring, J. Van Gemert, J. Geusebroek, and A. Smeulders. The challenge problem for automated detection of 101 semantic concepts in multimedia. In *Proceedings of the 14th international conference on Multimedia*, pages 421–430. ACM, 2006.

[12] M. Sullivan and M. Shah. Action mach: Maximum average correlation height filter for action recognition. In *CVPR*. IEEE, 2008.

[13] X. Wu, D. Xu, L. Duan, and J. Luo. Action recognition using context and appearance distribution features. In *CVPR*, pages 489–496. IEEE, 2011.

[14] J. Yang, R. Yan, and A. Hauptmann. Cross-domain video concept detection using adaptive svms. In *Pro-*

ceedings of the 15th international conference on Multimedia, pages 188–197. ACM, 2007.