SEMI-AUTOMATED VIDEO LOGGING BY INCREMENTAL AND TRANSFER LEARNING

Jongdae Kim and John Collomosse

Centre for Vision Speech and Signal Processing (CVSSP)
University of Surrey
Guildford, United Kingdom.

ABSTRACT

We describe a semi-automatic video logging system, capable of annotating frames with semantic metadata describing the objects present. The system learns by visual examples provided interactively by the logging operator, which are learned incrementally to provide increased automation over time. Transfer learning is initially used to bootstrap the system using relevant visual examples from ImageNet. We adapt the hard-assignment Bag of Word strategy for object recognition to our interactive use context, showing transfer learning to significantly reduce the degree of interaction required.

1. INTRODUCTION

Broadcast production is a heavy user of digital video, capturing substantial volumes of footage ("rushes") that often run to the order of days for a single hour long episode. The ability to efficiently annotate rushes with semantic metadata describing content (e.g. objects present) is essential to making sense of footage downstream in production. However semantic annotation of footage ("logging") is currently a labour intensive manual process, usually performed on-set in real-time due to the sheer volume of data. Although paper-based logging is prevalent, the industry is moving towards touch-screen loggers enable operators to tag video with a set of pre-configured labels. There is significant potential for Computer Vision to reduce the logging burden further through automatic recognition of objects in the real-time video stream.

In this paper we describe a semi-automatic system for logging video in real-time using a touch-screen interface. Our system incrementally learns the visual appearance of objects over time, using the interactive tagging actions of the logging operator. After a short period of time (a couple of hundred tags) the system is able to shoulder the logging burden, suggesting appropriate tags automatically.

As with existing touch-screen solutions, our system requires the specification of a pre-defined set of tags (object categories) prior to starting the video logging process. Each keyword is combined with the first few user-tagged examples of its respective object, and this information used to mine the ImageNet ontology for relevant visual examples of the object. This transfer learning process results in visual classifier, in the form of an Adaptive Support Vector Machine (A-SVM) for each object category. These classifiers are incrementally refined over time, as the user tags additional visual examples.

1.1. Related Work

Object recognition is a fundamental Computer Vision challenge. Contemporary approaches adopt a codebook approach, in which visual features (typically gradient-based [1, 2]) are detected and quantized into visual words [3]. Different object categories give rise to different frequencies of visual word occurrence. Supervised classifiers e.g support vector machines (SVMs) may be trained on this signal to discriminate between object categories. This strategy has been shown to scale to tens of object categories, but requires expensive training prior to an expensive optimization to train the SVM. We adopt incremental, rather than batch, training and use auxiliary sources to reduce user interaction.

Incremental (online) learning is often important to real world applications because complete data set is not always available at once. Early examples include the PAC (Probably Approximately Correct) learning of Anlguin *et al.* [4]. Generative incremental [5] has been introduced but still discriminative methods such as SVMs can achieve higher accuracy than generative model approach. The C&P algorithm [6]for SVMs offers exact solutions for online learning, maintaining the Kuhn-Tucker condition when training data is added or removed. Unfortunately C&P requires access to all historic training data, which is infeasible in our video annotation scenario. Bordes *et al.* proposed an approximate algorithm that optimizes the solution through the online sequential minimal optimization (SMO) algorithm [7]. Its efficiency in updating the SVM without a data storage overheads, coupled with comparable performance to batch trained SVMs, makes Bordes *et al* an attractive basis for our system.

Cross-domain (transfer) learning examples the training of a classifier using examples from a different context. Yang *et al* introduced the Adaptive SVM (A-SVM) [8], and initial experiments applying this to object detection have been reported by Tommasi *et al*. In our work we combine the A-SVM with incremental learning to train our object classifier using both user-supplied examples, and using creative commons imagery appearing within the ImageNet [9] ontology.

2. ONLINE OBJECT RECOGNITION

Our video annotation system is based upon a hard-assignment Bag of Visual Words (BoVW) framework. SIFT features [1] are computed from 20x20 overlapping patches densely sam-

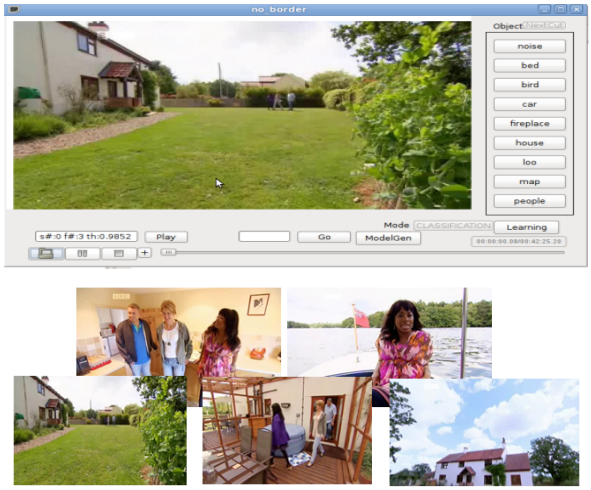

Fig. 1. Top. User interface of our semi-automated video logging system. Bottom: Representative frames from Escape to the Country, used for our evaluation. The video provides few uniform images of map, it can provide high performance

pled from video frames using the VLFeat [10] library. Features are quantized against a 2000 codeword dictionary constructed *a priori* from representative video frames. Codeword occurrence is counted across patches sampled from the frame, and normalised frequency histogram produced. The histogram is then passed to a bank of SVM classifiers to determine which objects, if any, are present. We adopt a one-vs-all approach to classification, with independently trained SVMs yielding a probability for the presence of each object class (frequently multiple objects of interest will be present in a frame). We employ both incremental and transfer learning extensions within our SVMs, as we now explain.

2.1. Adaptive SVMs for Transfer Learning

Our classifiers incrementally learn from user-tagged visual examples (subsec. 2.2). In practice a couple of hundred examples would be required before the classifier reaches optimal performance. We significantly reduce this latency (and improve overall accuracy) by boot-strapping the classifier using examples sampled from a visual ontology (ImageNet [9]).

We assume *a priori* knowledge of object category names, and use these at start-up and in combination with a few initial visual examples provided by the user, to mine ImageNet for relevant training examples. In practice we obtain the top 40 positive examples for each category from ImageNet using a GIST-based distance proposed by Ferrari *et al.*[11]. Incorporating user tagged examples to drive this distance is critical; e.g. in the ImageNet syn-set "car" not all cars may resemble the car being tagged within our video. Negative examples are also obtained from ImageNet by random sampling.

Adaptive SVMs (A-SVMs) [12] enable the combination of multiple SVMs such that an SVM trained using data from one domain (e.g. the ImageNet corpus) may be used to aid the classification decisions of an SVM trained using data from another domain (e.g visual examples from our video stream). For a given category we train two RBF-kernel SVMs, one (f^s) in a single batch process over the sampled ImageNet data (D^s), and another (f^t) incrementally on the user-tagged visual examples (D^t) (we discuss training of f^t in subsec. 2.2).

We use two examples per categories from Video stream to obtain new classifier f, each a pair$(x_i, y_i)_{i=\{1,2\}}$ where x_i is the image descriptor (BoVW histogram computed using the codebook established during pre-processing) and $y_i = \{-1, +1\}$ is binary label indicating a positive or negative training exemplar. The A-SVM modified the classical SVM by introducing a delta function defined as $\triangle f(x_i) = w^T \phi(x_i)$ where function ϕ maps vector x_i to feature vector $\phi(x_i)$ and w is a parameter optimized during training to balance the decisions of the two classifiers (f^s and f^t). The A-SVM combines the two classifiers and vectors as follows:

$$\begin{aligned} f(x) &= a_1 f^s(x) + a_2 f^t(x) + \triangle f(x_i) \\ &= a_1 f^s(x) + a_2 f^t(x) + w^T \phi(x_i). \end{aligned} \tag{1}$$

where a_1 and a_2 are normalised weights of source classifiers f^s and f^t that may be user defined (by we take as equal). We learn w to solve eq.1 since f^s trained by a batch algorithm and f^t trained by online algorithm are already provided. The objective function to learn w is given by:

$$\min_w \frac{1}{2}\|w\|^2 + \mathbf{C}\sum_{i=1}^{N}\xi_i \tag{2}$$

$$\text{s.t}\, \xi_i \geq 0, \tag{3}$$

$$y_i\{a_1 f^s(x) + a_2 f^t(x)\} + y_i w^T \phi(x_i) > 1 - \xi_i.$$

where ξ_i is the hinge error function of $f(x_i)$, C controls the weight of the classification error function and N is the number of data vectors. Variable w in the regularization term $\|w\|^2$ is the parameters of $\triangle f(x)$ rather than the final combined classifier $f(x)$. This approach provides higher accuracy than a simple linear combination of the two SVMs or batch based ASVM algorithm although the experiment result is not presented here due to page limitation.

2.2. Incremental training of the SVM

The performance boost afforded by knowledge transfer in the A-SVM is attractive, however A-SVMs are classically composed of batch trained SVMs and cannot be updated without retraining. To avoid this we exchange the batch trained SVM $f^s(t)$ for an incrementally trained SVM adopting the LASVM algorithm for online learning [7]. The LASVM is an effective online algorithm which provides fast training, using little memory (no historical data points need be retained) and as we later show produces similar performance to classical batch based SVM training.

The LASVM relies on the sequential minimal optimization(SMO) calculating an "τ approximate solution" to solve the optimization problem. The detail of SMO algorithm can be found in [13].

We begin by creating a set of binary classifiers, using the one vs all strategy to solve our multi-category object classification task. When a user tags a video frame as contain-

ing a particular object category, the descriptor (BoVW histogram) computed from that frame is entered into the relevant LASVM classifier as positive exemplar and into the others as a negative exemplar. The brief description below outlines how the an LASVM is updated for a given example (x, y) where x is the descriptor and $y = \{-1, 1\}$ indicates negative or positive class membership for that classifier.

The online learning of LASVM is defined by PROCESS and REPROCESS working on pairs of examples. The PROCESS operation involves a new given vector x and one vector from current support vectors *S*. The x might be a support vector if the coefficient of x is not zero after PROCESS operation, the example become a member of current vectors *S*. The REPROCESS search operation is dealing with two examples of the current support vectors *S* in the current kernel expansion. The operation changes the coefficient of one or both vectors to zero which are not candidates of the current set of support vectors anymore.

The final result *S* provides support vectors and coefficients for a new classifier $f(x)$ which is used as f^t within the A-SVMs. To perform the pair of processing operations only the new set of exemplars and existing set of support vectors need be retained, obviating the need to retain increasing volumes of descriptor histories over many hundreds of video frames.

3. RESULTS AND DISCUSSION

We evaluated our video annotation system using a 40 minute episode of "Escape to the country" (Boundless); a broadcast television show on the topic of property. Eight object categories were evaluated: Bed, Bird, Car, Fireplace, House, Bathroom, Map, People. Shot detection was applied to the sequences, and 1-2 frames per shot extracted to create a dataset of 567 frames each of which was manually annotated to indicate which objects categories were present. The data was then split chronologically into a training set (285) and test set (282). Such shot-centric sampling was adopted since ground-truth markup of the entire episode is intractable, and sampling a couple of frames per shot removes bias from lengthy shots containing footage (e.g. of a map) that is less challenging to recognise.

The Mean Average Precision (MAP) over a given frame was computed as the mean of the average precision of each trained classifier over that test frame. The MAP of the system is reported as the average MAP across all 282 test frames.

3.1. Incremental Learning

We trained our system incrementally, measuring MAP over all categories after exposure to visual examples present in 2, 10 , 50 and 285 training frames. We compared these results to a classical offline object recognition system trained via a one-off batch process over tags accumulated from the same time intervals. Features, codeword dictionary and training parameters of the non-linear SVM were constant between all systems and trials. Figure 2 illustrates comparable performance of the two systems, indicating that an incrementally trained classifier offers no performance disadvantage over a classifier trained once with sight of all data. In these examples no transfer learning was applied to boot-strap the classifier. Overall per-category performance is summarised as average precision (AP) in Table 1.

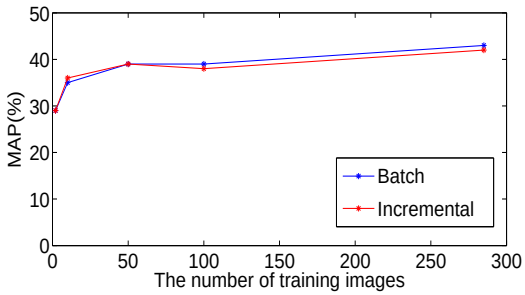

Fig. 2. MAP is maintained as volume of training data increases, comparing batch vs. incremental training.

Table 1. Comparison of online(O) / offline(F) SVM (% AP)

Car		House		Bird		Bed		Firepc		Bathrm		Map		People	
F	O	F	O	F	O	F	O	F	O	F	O	F	O	F	O
4	4	41	41	3	3	13	13	10	10	24	24	100	100	40	40
16	16	38	39	21	22	14	15	15	15	25	26	100	100	56	57
7	8	48	51	29	29	13	13	10	11	39	40	100	100	66	67
15	13	58	57	20	19	17	15	10	10	36	36	100	100	62	61
13	12	67	66	29	30	15	14	15	14	39	38	100	100	66	67

3.2. Transfer Learning

We compared the performance of our incrementally trained classifier running both with the transfer learning component (A-SVMs) and without (incrementally trained SVMs). Again, MAP was plotted after exposure to visual examples provided in 2, 10 , 50 and 285 training frames as shown in the Figure 3. As expected with very few visual examples, the transfer learning system (TF) significantly out-performs the system without (NTF). The rate of learning is significantly improved in TF (confirming [14]), and leads to a marginal performance increase over NTF given the same set of visual examples. Corresponding average precisions (AP) for each category are given in Tables 2 and 3.

3.3. Timing Analysis

We ran timing analysis on the training and test components of the full system (incorporating both incremental and transfer learning). The code was implemented in single-threaded C/C++ on a Pentium 4 processor running at 3.6Ghz.

The prediction (classification) time per A-SVM was measured at $0.28 - 0.75ms$ per frame, yielding an average performance over all classifiers of 0.54ms per frame (20 fps). However the SIFT feature extraction step is significantly more expensive (400ms, 2.5fps). Switching to multi-threaded code, or a comparable descriptor amenable to GPU implementation (e.g. SURF), are promising ways to reduce this overhead. On

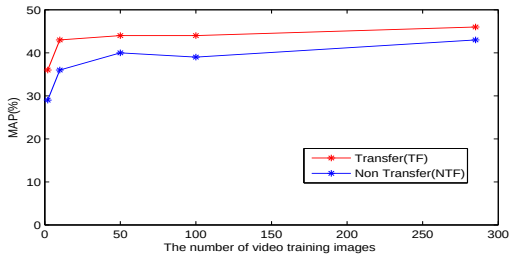

Fig. 3. Transfer learning using ImageNet improves both rate of learning and performance. The transfer learning rapidly reaches the final performance while classifiers of non-transfer learning are being trained.

Table 2. TF vs. NTF with 2 training examples/category (% AP)

	Bed	Bird	Car	Firepc	House	Bathrm	Map	People
NTF	5	3	4	5	41	24	100	30
TF	10	32	5	12	58	30	100	54

Table 3. TF vs. NTF with all training data (% AP)

	Bed	Bird	Car	Firepc	House	Bathrm	Map	people
NTF	14	30	12	14	66	38	100	67
TF	14	57	9	14	66	39	100	69

the other hand, not every frame of video need be annotated in our application to enable useful semantic search/retrieval later in the production pipeline. The incremental learning (update) time per A-SVM was measured at 150ms (7fps) per additional visual example supplied. This method is significantly faster than user input could be provided in use, and so is acceptable for our application.

4. CONCLUSION

We have demonstrated an online video logging solution using incrementally trainable classifiers performing object recognition. Our novel contribution is in the fusion of an Adaptive SVM for transfer learning with incrementally trainable SVM, yielding a system capable of learning object visual appearance from only tens of user tagged examples. We have shown that transfer learning in this context enables reduction of the number of training interactions by one order of magnitude, and a marginal increase in the overall classification performance across our test categories. We have also shown that incremental learning does not reduce overall accuracy when compared against batch training approaches.

Large scale object recognition remains an open challenge in Computer Vision. As the number of object classes increases, solutions will be required to reduce the manual training burden associated with supervised classification. Our combination of incrementally trainable classifiers with one-shot transfer learning from ImageNet is one step toward this goal. Our next step will be to mitigate the linear growth in number of SVMs with object category count inherent in our one versus all strategy. To this end we intend to draw upon ImageNet to build a hierarchy of semantic concepts to reduce the run-time complexity of classification.

5. REFERENCES

[1] David G. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, vol. 60, pp. 91–110, 2004.

[2] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," *CVPR*, vol. 1, pp. 886–893, 2005.

[3] Sivic and Zisserman, "Video Google: a text retrieval approach to object matching in videos," in *IEEE Conference*, 2003.

[4] Dana Angluin, "Queries and Concept Learning," *Mach. Learn.*, vol. 2, no. 4, pp. 319–342, Apr. 1988.

[5] Li Fei-Fei, Rob Fergus, and Pietro Perona, "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories," *Comput. Vis. Image Underst.*, vol. 106, no. 1, pp. 59–70, Apr. 2007.

[6] Gert Cauwenberghs and Tomaso Poggio, "Incremental and Decremental Support Vector Machine Learning," in *NIPS*, 2000, pp. 409–415.

[7] Antoine Bordes, Seyda Ertekin, Jason Weston, and Léon Bottou, "Fast kernel classifiers with online and active learning," *JMLR*, vol. 6, pp. 1579–1619, 2005.

[8] Jun Yang, Rong Yan, and Alexander G. Hauptmann, "Adapting svm classifiers to data with shifted distributions," *IEEE International Conference on Data Mining Workshops*, 2007.

[9] Jia Deng, Wei Dong, R. Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *CVPR*, 2009, pp. 248–255.

[10] Andrea Vedaldi and Brian Fulkerson, "Vlfeat: an open and portable library of computer vision algorithms," in *the international conference on Multimedia*, 2010.

[11] T. Deselaers and V. Ferrari, "Visual and semantic similarity in imagenet," in *CVPR*, 2011, pp. 1777–1784.

[12] Jun Yang, Rong Yan, and Alexander G. Hauptmann, *Cross-domain video concept detection using adaptive svms*, MULTIMEDIA. ACM, 2007.

[13] John C. Platt, *Advances in kernel methods*, MIT Press, Cambridge, MA, USA, 1999.

[14] T. Tommasi, F. Orabona, and B. Caputo, "Safety in numbers: Learning categories from few examples with multi model knowledge transfer," in *CVPR*, 2010.