VIDEO CATEGORIZATION USING OBJECT OF INTEREST DETECTION

Adarsh Kowdle\textsuperscript{1}, Kuo-Wei Chang\textsuperscript{2}, Tsuhan Chen\textsuperscript{1}

\textsuperscript{1}Cornell University, NY, USA. \textsuperscript{2}Chunghwa Telecom Co., Ltd, Taiwan.

ABSTRACT

Object of Interest (OOI) detection has been widely used in many recent works in video analysis, especially in video similarity and video retrieval. In this paper, we describe a generic video classification algorithm using object of interest detection. We use online user-submitted videos and aim to categorize the videos into six broad categories: hot star, news, anime, pets, sports, and commercials. We show through our experiments that, detecting and describing the object of interest improves the video classification accuracy by about 10 percentage points.

Index Terms— Video classification, Object of Interest detection

1. INTRODUCTION

In this paper, we describe an internet video classification algorithm based on Object of Interest (OOI) detection. The general problem of internet video classification is known to be hard, due to the lack of constraints in the user-submitted videos [16]. However, we show that discovering the OOI in the video can considerably aid this task and improve the performance of the classifier.

Detecting the OOI in a series of images is already a mature technique in surveillance and object segmentation [6]. Recently, this method was extended from a group of images to videos, for tasks like video segmentation, object tracking, video retrieval, and key frame selection [8–10]. In this paper, we use object of interest detection to aid internet video classification.

Video classification has been an active research area for many years. These algorithms can be broadly classified into two types of video classification algorithms. The first type of classifier is a category-specific video classifier, which classifies videos from a particular category, such as sports, into categories, such as tennis, baseball [5, 17]. Liu et al. [11] showed an approach to categorize human actions by information maximization however, in typical user uploaded videos (like we consider) we would want to define more generic video categories than well-defined human actions. OOI has been shown to improve the performance of video retrieval [10], however, this can also be considered a category-specific retrieval as the categories chosen were specific like giraffe, helicopter, space shuttle, etc. The second type of classifier is a generic video classifier, which classifies the videos into generic categories, such as sports, commercials, news, animation, etc [16, 18]. This paper differs from [10] and focuses on the second type of classification, a generic video classifier.

There are a lot of approaches one can consider for classifying videos. SVM (Support vector machine) and HMM (Hidden Markov model) are examples of model-based classifiers [5, 13, 17]. However, recent work on image classification has shown that nearest neighbor based classifiers serve as fast classifiers which can provide good performance even with large datasets [4]. In this paper, we propose a nearest neighbor based generic internet video classifier using object of interest detection.

The rest of this paper is structured as follows. In Section 2, we briefly review the OOI detection algorithm. The generic video classifier is described in Section 3. The results and discussions are provided in Section 4 followed by the conclusions in Section 5.

2. OBJECT OF INTEREST DETECTION

Object of interest detection has been successfully used in many recent works. There are many approaches to extract the object of interest [7, 9, 14]. The OOI detection algorithm we use is based on the work by Liu et al. [9]. We represent the OOI detection as a probabilistic model which combines appearance and spatial distribution of the object of interest, which are summarized below.

2.1. Appearance and spatial modeling

We start by finding a number of patches in every frame of the video to generate visual words which help describe the appearance of the object of interest. The patches are obtained by using the Maximally Stable Extremal Regions (MSER) operator [15]. SIFT features are then extracted from these patches [12] which yield a 128 dimensional descriptor for each MSER patch. The descriptors collected from all the frames are vector quantized using k-means clustering resulting into \( C \) cluster centers (\( C = 50 \), in our case) to form the codebook of visual words, \( \{ w_1, w_2, \ldots, w_C \} \). Each MSER is now described using discrete visual words instead of the
continuous SIFT descriptor.

**Appearance modeling**

Consider the frames of the video denoted by \( \{ d_1, \ldots, d_N \} \), we define the hidden variables \( z_i(k) \) to indicate if the \( i^{th} \) MSER from frame \( k \) came from the object of interest (OOI) or the background (bg), which we wish to estimate. We model the appearance using two conditional probabilities, \( P(z_i(k)|d_k) \) and \( P(w_j|z_i(k)) \) for each MSER. \( P(z_i(k) = \text{ooi}|d_k) \) indicates how likely the \( i^{th} \) MSER in frame \( k \) originates from the object of interest. \( P(w_j|z_i(k) = \text{ooi}) \) indicates how likely the \( i^{th} \) MSER which originated from the OOI in frame \( k \) has an appearance corresponding to the visual word \( w_j \). \( P(z_i(k) = \text{bg}|d_k) \) and \( P(w_j|z_i(k) = \text{bg}) \) are defined analogously for the background.

**Spatial modeling**

We denote the position of the \( i^{th} \) MSER from frame \( k \), as \( r_i(k) \) and the corresponding hidden variable as \( z_i(k) \) along similar lines as appearance modeling. We describe the spatial distribution as \( P(r_i(k)|d_k, z_i(k) = \text{ooi}) \) (written in simplifies notation as \( P(r|d, \text{ooi}) \)) and \( P(P(r_i(k)|d_k, z_i(k) = \text{bg}) \) to describe how the object of interest and background clutter are distributed in the frame \( k \). Incorporating this model makes sure that the temporal changes in the OOI enters the probabilistic model and influences the model.

The probabilistic model that combines the appearance and spatial information is described using the following joint distribution. (subscripts dropped for legibility)

\[
P(d, w, r, z) = P(r|d, z)P(w|z)P(z|d)P(d) \tag{1}
\]

Now, the posterior probability \( P(\text{ooi} | d, w, r) \), calculated as follows:

\[
P(\text{ooi} | d, w) = c * P(z|d)P(w|\text{ooi})P(r|d, \text{ooi}) \tag{2}
\]

where, \( c \) is a normalizing constant to make it a probability mass function.

**2.2. OOI Algorithm**

**Maximum likelihood parameter estimation**

The distributions \( P(z_i(k)|d_k) \), \( P(w_j|z_i(k)) \), \( P(r_i(k)|d_k, z_i(k)) \) and hence the posterior, \( P(z_i(k)|d_k, w_j, r_i(k)) \) are estimated using the Expectation-Maximization algorithm as described below:

\[
E - \text{Step:} \quad P(z_i(k)|d_k, w_j, r_i(k)) = \frac{c_1 * P(z_i(k)|d_k)P(w_j|z_i(k))P(r_i(k)|d_k, z_i(k))}{
\sum_j \sum_i c_2 \sum_k \sum_i \co_{kji} P(z_i(k)|d_k, w_j, r_i(k))}
\]

\[
M - \text{Step:} \quad P(z_i(k)|d_k) = \frac{c_2 * \sum_j \sum_k \co_{kji} P(z_i(k)|d_k, w_j, r_i(k))}{
\sum_j \sum_i \co_{kji}} \tag{4}
\]

\[
P(w_j|z_i(k)) = \frac{c_3 * \sum_k \sum_i \co_{kji} P(z_i(k)|d_k, w_j, r_i(k))}{
\sum_i \co_{kji}} \tag{5}
\]

\[
P(d_k) = c_4 * \sum_j \sum_i \co_{kji} \tag{6}
\]

\[
P(r_i(k)|d_k, z_i(k)) \text{ using the particle filter.} \tag{7}
\]

where, \( \co_{kji} = co(d_k, w_j, r_i(k)) \) represents an element in the co-occurrence matrix which gives a count of the number of occurrences of word \( w_j \) at position \( r_i(k) \) in frame \( d_k \). \( \{ c_1, \ldots, c_4 \} \) are the normalization constants to make the functions probability mass functions.

We define the ‘posterior map’, or P-Map as an image which stores the posterior probability \( P(\text{ooi} | d, w, r) \) which is updated using the EM approach for every . Similarly, a
‘location map’, or L-Map stores the probability $P(r|d, z_{ooi})$. A particle filter is used to obtain the L-Map. The particles used for the particle filter for each frame are represented by the position, scale and velocity as:

$$x^{(i)} = (pos_h^{(i)}, pos_v^{(i)}, scale_h^{(i)}, scale_v^{(i)}, vel_h^{(i)}, vel_v^{(i)})^T$$ (8)

The input to the particle filter is the P-Map from the previous iteration. In each step the particle filter cleans the P-Map to give the L-Map. The P-Map at the end of a few iterations of the EM approach (we use 20 iterations), gives the posterior distribution of the OOI in each frame. This posterior distribution is thresholded and bound to extract the object of interest.

3. GENERIC VIDEO CLASSIFIER

In this section, we describe our nearest-neighbor based generic video classifier using object of interest detection.

The MSER patches and SIFT descriptors described in the Section 2.1 are reused here. The SIFT descriptors collected from the frames of all the videos in the database are vector quantized using k-means clustering into $K$ cluster centers ($K = 1000$, in this case) to form the codebook of visual words, $\{W_1, W_2, \ldots, W_K\}$. Note here that, the codebook generated here is a global codebook (i.e. across videos), but, the codebook in Section 2.1 was generated for each video. This is important because in Section 2.1 we were focusing on extracting the OOI within a video sequence however, for categorization we need a global codebook across all video sequences to help describe each video as a whole. We experimented with other features like spatio-temporal features [18] however, our experiments showed that we would not gain much with those features because of the randomness in the motion of objects in the internet videos we use.

A bag of words model is used to perform the classification. Each descriptor is described using the nearest visual word. These words are accumulated to obtain a codeword histogram for each video. The histogram is normalized by the number of frames in the video. This allows us to define the distance between two videos as the Euclidean distance between the corresponding normalized histograms. For a new query video, the system evaluates the codeword histogram and finds the nearest neighbor in the database. The query video is classified into the same category as the category of the nearest neighbor in the database.

The video classifier defined above is similar to the system described by Schindler et al. [16]. We use this as the baseline in our work. We develop our proposed generic video classifier incorporating the detected OOI into this approach. We build the histogram by giving importance to the object of interest instead of the whole frame. The histogram is normalized by the number of frames that contain the OOI. This training phase incorporating the object of interest detection is illustrated in Fig 1. For a new query video, the system first detects the OOI region and then obtains the codeword histogram. Finally, the system finds the nearest neighbor in the database and classifies the query video. The testing phase is illustrated in Fig 2.

4. RESULTS

In this section, we first describe the dataset we collected, followed by results and discussion.
Video Categorization using OOI (Average: 34.8%)

Performance with an average accuracy of 34.8% video classifier using OOI detection resulted in a better performance important regions of the video. Thus, improving the average accuracy of a generic video classifier from 27.3% to 34.8%.

Dataset

Our dataset contains 400 videos and is collected from community websites Xuite [2], YouTube [3] and blinkx [1]. It was manually labeled into 6 categories: anime, news, pets, hot stars, sports and commercials. Some frames from sample videos from the dataset are shown in Fig 4.

Experiments and results

We use leave-one-out cross-validation (LOOCV) for all our experiments. We first evaluate the performance of our baseline generic video classifier on the dataset as described in Section 3 which is similar to [16]. The average accuracy of this classifier was about 27.3%. The confusion matrix is shown in Fig 4a. Although the dataset we use is different from the prior works, our baseline result is comparable to prior work [16]. The same setup was used and the object of interest detection is now incorporated into the system. Some sample frames showing successful detection of the object of interest is now incorporated into the system. Some sample frames showing successful detection of the object of interest with our dataset, are shown in Fig 4. The generic video classifier using OOI detection resulted in a better performance with an average accuracy of 34.8%. The confusion matrices for the proposed classifier is shown in Fig 4b.

We note from the confusion matrix that, the accuracy of the class anime increases significantly on using the OOI, as compared to classes like news and hot stars. The main reason for this is that, using OOI can definitely boost the performance of the classifier if the OOI is well defined, such as, Snoopy or Garfield in the Anime category. However, with categories such as news and hot-stars, the OOI are always people and so information about the OOI in this case would not be discriminative between the categories especially because even within each category the variance between the people (OOI) would be very large. However, on an average, we observe that using OOI can help boost the performance of the video classifier by about 8 percentage points.

Fig. 4: Confusion matrices for the 6 class generic video classifier.

5. CONCLUSIONS

In this paper, we describe a generic internet video categorization approach using object of interest detection. We show through our experiments on a large dataset that using the object of interest detection helps guide the video classifier towards important regions of the video. Thus, improving the average accuracy of a generic video classifier from 27.3% to 34.8%.

6. REFERENCES