Denoising face recognition via clustering techniques

Trung Vu, Sean Lee, Alexandre Lamy

Abstract

In this paper we address the problem of detecting and recognizing specific objects (such as the faces of a specific group of people) in real life videos. By using unsupervised clustering techniques, we are able to exploit the inter-frame relationships in the video to significantly improve the accuracy of basic object detection and recognition algorithms. This yields results which are also much more robust to frame switches, lengthy occlusions, and variable numbers of targeted objects leaving and reentering the frames (all common occurrences in real life videos) than standard tracking techniques.

1 Introduction

One of the most interesting and difficult problems in computer vision is that of recognizing and then tracking an entity of interest over time (i.e. in videos). This can be useful for surveillance of suspect individuals, analysis of video data (music videos, concerts, sports games, etc.), or in building robots that interact with their environment.

While tracking is a topic that has been extensively studied (see [1]), it is usually done so in isolation from recognition. To track the object of interest, the leading tracking algorithms mainly rely on properties of videos, notably that tracking target(s) will move by small amounts at a time. This allows the algorithms to detect/infer the object(s) positions from the previous ones by focusing on the region around the previously known position. While, these methods can work remarkably well on a clean video, such techniques have little hope when faced with long occlusions or frequent "frame switches" as these will cause a violation of the assumptions that the algorithms are based on.

Alternatively, object detection techniques could be applied frame by frame. This would result in a much more robust result since the algorithms make no assumptions concerning the relations between different frames. Hence, occlusions and frame switches pose no issue whatsoever. Unfortunately, even the best of these methods result in relatively noisy or inaccurate results when compared with the tracking algorithms. This can be attributed to the fact that the detection algorithms make no use of the wealth of information provided by the inter frame relationships (notably that most objects will usually not move by a large distance).

In this paper we propose a novel post-processing method, based on unsupervised clustering techniques, to "denoise" results obtained from object detection algorithms by exploiting inter-frame relationships. The result is a technique that is more robust to occlusions and frame switches than the standard tracking techniques and more accurate (less noisy) than the results obtained by naively applying object detection algorithms. We also have the added benefit of being able to easily do object recognition at the same time as detection. Our method will not only reduce the noise in the detection error (bad bounding boxes) but also in the classification error (bad labels).

2 Standard tracking methods and issues

Our problem, that of detecting objects throughout a video, or variations of it are often solved via tracking techniques. As mentioned above, the core idea is to exploit inter-frame relations, notably the assumption that most objects will not move by much between frames, to track or follow the

various target objects. We quickly summarize the main tracking algorithms used in practice and show that they rely on core assumptions which are violated when faced with real life videos which contain frame switches, long occlusions, and various target objects coming in and out of the frames in variable number. In the presence of these realities, we show that these tracking algorithms are unusable.

As mentioned in [1], the most commonly used tracking algorithms are OLB [4], IVT [5], MIL [2], L1 [6], and TLD [7]. We give a quick summary of each and outline their main issue in our context.

OLB or tracking via on-line boosting is a tracking method described in [4] which takes the most basic approach to tracking and use Adaboost to achieve better performance. The core basic approach, which most of these other methods also include in some way, is to use a classifier to evaluate many possible regions surrounding the previously known location of the object. This creates a confidence map which is then analyzed to find the most probable current position of the object. Finally, the classifier is retrained on the new position as well as the previous ones. The addition of OLB is to have many different trackers (classifiers), view them as weak learners, and use Adaboost to obtain a classifier with a much better performance by aggregating the weak learners. Clearly, this method has the drawback we mentioned earlier. Since it retrains every frame, long occlusions can completely wreck the classifier. Even more problematically, the tracker needs to be given the initial position of the object. This means that handling frame switches or new objects coming in and out will be very problematic. Even, if adhoc techniques are used to detect frame switches, the algorithm would still need to somehow get the objects' new initial position. Using object detection techniques to find these is dangerous since these techniques are fairly noisy and giving a bad initial position would be disastrous (since all the subsequent positions will be found based on that first position).

Incremental learning for robust visual tracking is a technique described in [5] which is more robust to changes in the objects' appearance or illumination. By efficiently doing only incremental updates on a learned eigenbasis as new observations come in, the resulting classifier is more robust to unexpected changes in appearance and lighting. However, this does nothing to build robustness against the problems we are interested in in our setting. As for OLB, the algorithm works by looking at locations close to the previous position. The algorithm also requires being given the initial position of the object being tracked, which again can be very problematic.

Another method, robust visual tracking via ℓ_1 minimization ([6]), does try to deal with the problem of occlusions. The idea is to have a set of templates. Some, target templates, contain the target object. Others, trivial templates, are simply templates with a single nonzero element. Then the idea is to try to represent the target candidate as a sparse linear combination of templates. Clean versions of the target object should be sparsely captured by a few target templates while occlusions should be captured by a sparse number of trivial templates. This allows occlusions to be dealt with naturally and without damaging the classifier (the target templates won't be corrupted nor added to when occlusions happen). The method also uses a particle filter technique to estimate and propagate the posterior distribution over the motion of the object over time. This allows selecting the target candidate that is then matched to templates. However, this means that large movements as caused in frame switches can cause massive problems for this method. Having objects go in and out of the frame can also be a major issue as initial templates need to be provided each time.

One of the most popular methods for tracking is MILtrack [2]. This method uses multiple instance learning to track the object. To understand why we need multiple instance learning, consider the following approach for tracking. Given a location of the object in the last frame, we want to estimate the next location of the object in the current frame. A way to do this is to use image patches of the object itself and a neighborhood around it as positive examples, and image patches of the background surrounding the object as negative examples. An issue that arises here is that it is ambiguous how to define an object: a precise bounding box seems to restrictive, and furthermore, if an object is a person, then they might be sitting, standing, etc. Multiple instance learning provides the solution to this problem by providing the positive examples as a bag of image patches instead of a single image patch. Then we can compute features for the patches and then use a classifier to learn the object. In the case of the [2], the classifier was a variant of AdaBoost used over Haar-like features. An online variant of the MILtrack algorithm can be found in [3].

Another method for tracking is TLD (Tracking-Learning-Detection) [7]. This method combines both tracking and detection into one single, unified framework. In this model, both tracking and detection output their predicted bounding boxes separately, and an integrator updates the location of

the object based on predictions from both of these models. Furthermore, at each step, the model also utilizes a P-expert, which identifies false negatives, and an N-expert, which identifies false positives. The P-expert relies on the assumption that location does not change much from frame to frame to identify false negatives, while the N-expert relies on the assumption that an object only appears at a single location at a time to identify false positives. The detection model is updated at each time step according to the examples generated by the P-expert and the N-expert.

3 Standard object detection and recognition methods

Traditionally, the task of object detection has been carried out training discriminative models (AdaBoost or SVM) over manually-engineered features (Histogram of Oriented Gradients (HOG) [8] or Haar-like [9]). These methods have achieved certain successes, but are often computationally heavy and inaccurate.

Recently, methods using neural networks have emerged. These methods apply neural networks, which are computationally expensive to train but have incredible expressive capabilities, to object detection task. Methods such as R-CNN [10] or YOLO [11] have produced state-of-the-art result in object detection.

Our paper will focus on face detection, and will employ a popular recent method in face detection [12]. This method uses a multi-task CNN that splits the detection task into 3 phases: the first phase comes up with proposed face regions, the second phase refines the suggestions from the first phase, and the third phase detects facial landmarks (eyes, noses, lips, etc.). The multi-task component of these neural networks stem from the fact that all three networks are forced to output 3 things: 1) bounding boxes, 2) whether those bounding boxes are faces, and 3) the facial landmarks on the detected face.

This object detection method provides another method to face-tracking. One can just detect the faces frame-by-frame. This does not force any assumptions on the model (as opposed to tracking model, where the object is often assumed to be visible at all times or move "slowly"). The issue, however, is that we need to connect these noisy objects detected on a frame-by-frame basis into a coherent entity that exists intertemporally.

4 Our method: using clustering to exploit video structure in order to denoise detection and recognition methods

5 Experiments and Results

Acknowledgments

References

References

[1] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Online object tracking: A benchmark. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013.

[2] Helmut Grabner, Michael Grabner, and Horst Bischof. Real-time tracking via on-line boosting. In *Bmvc*, volume 1, page 6, 2006.

[3] David A Ross, Jongwoo Lim, Ruei-Sung Lin, and Ming-Hsuan Yang. Incremental learning for robust visual tracking. *International journal of computer vision*, 77(1-3):125–141, 2008.

[4] Cha Zhang, John C. Platt, and Paul A. Viola. Multiple instance boosting for object detection. In Y. Weiss, B. Schölkopf, and J. C. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 1417–1424. MIT Press, 2006.

[5] Xue Mei and Haibin Ling. Robust visual tracking using l1 minimization. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1436–1443. IEEE, 2009.

[6] Zdenek Kalal, Jiri Matas, and Krystian Mikolajczyk. P-n learning: Bootstrapping binary classifiers by structural constraints. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 49–56. IEEE, 2010.

[7] Boris Babenko, Ming-Hsuan Yang, and Serge Belongie. Robust object tracking with online multiple instance learning. *IEEE transactions on pattern analysis and machine intelligence*, 33(8):1619–1632, 2011.

[8] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

[9] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–I. IEEE, 2001.

[10] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[11] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[12] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, 2016.