Video understanding using part based object detection models

Vignesh Ramanathan
Stanford University
Stanford, CA-94305
vigneshr@stanford.edu

Kevin Tang (mentor)
Stanford University
Stanford, CA-94305
kdtang@cs.stanford.edu

Abstract

We will explore the use of event specific object detectors (part based models) in multimedia event detection. The challenge is to choose a well-trained object detector specific to the event videos. The presence of dataset bias would make an object detection model trained on an unrelated generic image database less useful for the video dataset in hand. Given such a generic object model, we propose an iterative method to build a more effective detector, trained only on frames from the training video dataset. The generic model, in combination with an optical flow based filtering method is used to extract objects with high confidence from training videos, which are then used to train a new object detection model. This model is again used to extract objects, used for training in the next iteration. The process is repeated to finally obtain a dataset specific model trained only on frames from training videos. The performance of the final model is evaluated on manually annotated frames form test videos. It is compared with the original object detector to show the gain of the proposed method. A group of such dataset specific object models corresponding to different objects can be used to build features from videos for high level vision tasks. Finally, in order to evaluate the new model in the context of video understanding, the two models are used to extract some simple features from a set of videos belonging to two event classes and corresponding Precision Recall curves are presented for the two models in this binary classification setup.

1. Introduction

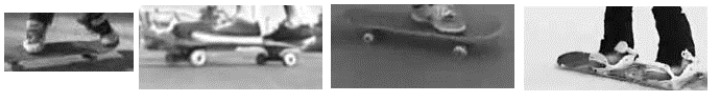
Video understanding aims to identify spatial and temporal patterns in a video to recognize the events captured by it. Given a set of pre-defined events, multimedia event detection identifies the occurrence of an event in a video-clip. This is akin to the fundamental challenge of object recognition in images. The difficulty of the event detection task arises from the huge interclass variation in camera view points, appearance of objects/ persons involved in the event, resolution, illumination, video quality etc.

In this project, we will focus on the task of event detection using event specific object detectors. In general, such detectors are a part of a larger framework, where the motion of the object is also identified in successive video frames and compared with corresponding motion in training videos [6, 8]. However, in this project we will restrict the analysis to tagging videos based only on detection of event specific objects. In particular, we propose a method to build an object detector which would perform well for a given video dataset.

An object detector trained on a generic image database like Imagenet [1] would not be effective on the video dataset, due to the presence of inherent dataset bias. For instance, in the case of detecting "skateboards" in skateboarding videos, it can be seen that the videos mostly contain frames showing people moving on skateboards. On the other hand, Imagenet skateboard images show skateboards from different views often occluded by other objects. Some sample "skateboard" images from Imagenet database and "skateboard" images segmented from Trecvid video sequence are shown in Fig. 1. The effect of such dataset bias has been explored in [10]. The paper has analyzed the performance of detectors trained on one dataset and tested on others. The performance was seen to degrade even for the two class classification problem. Hence, in order to achieve best results, we would like to train the object detector only on video frames from the training video dataset. However, it is impractical to manually annotate video frames, every time we are given a video dataset. Instead, we will use the object detector trained on a readily available annotated image dataset like [1] to build an object detector specific to the given video dataset. This object detector will be used to extract relevant object sequences from event videos and tag them according to the presence or absence of such object sequences.

A part based model for object detection was proposed in [3] and shown to achieve state-of-the-art results on the PASCAL VOC benchmarks [2]. [3] represents object classes as multi-scale models with deformable parts. We will use this

(a)



(b)

Figure 1. Sample "skateboard" images from (a) Imagenet database and (b) video frames from Trecvid video database. The difference in orientations and context surrounding the object can be seen in these images.

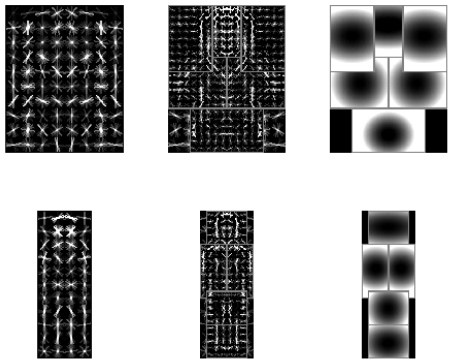


Figure 2. An illustration of the part based model for the object class "person". The two rows corespond to the two components of the model. The first colum shows the root filter (The HOG feature representation). The second column shows the different part filters of the model and the last column shows the deformation cost assocaited with the plasemnt of each part with respect to the root filter.

part based model obtained from [4] to detect event related objects from training videos and iteratively train the model with the segmented objects. This would improve the performance of the model for videos belonging to the event set. While detecting objects from videos for training, an optical filter[5] based filtering is applied in the temporal domain to ensure that only objects which are detected consistently in successive frames and with motion along the path predicted by optical flow are retained. This minimizes the chance of spurious detections. The final improved detector can be used to extract object (pertaining to a certain event) sequences from a test video.

The initial part based model for detecting event related objects in videos is trained with images obtained from ImageNet [1]. TRECVID [9] event kits is used for training and testing the proposed algorithm. Each event kit contains the definition and evidential description of the event. For a specific event, the event related objects are decided based on this evidential description.

It was shown in [7], that a bank of object detectors can be used to build powerful features from images for high level vision tasks like scene understanding. We extend this idea to find the utility of a set of dataset specific object models in the context of video understanding. In particular, we present results for binary event classification problem by considering a simple feature (developed along the lines of [7]) extracted from videos, using a set of two object detectors. We compare the results for object models trained on a generic database as well as the iteratively trained models.

2. Background

We have used the part based model [3] for object detection from target video frames. The current work also explores a strategy similar to [7], for event detection by using a bank of object detectors to identify relevant object sequences from videos. In addition to an object detector, we use the optical flow based tracking algorithm to deduce the motion of an object in successive frames. The part based model and optical flow algortithms are discussed in Sec. 2.1 and Sec. 2.2. We have also provided a brief description of the objectbank scheme for scene classification in Sec. 2.3.

2.1. Part based model for object detection

The part based model is a deformable model which makes uses of a root filter and a set of part filters to identify objects. At a given scale and position in an image, the detection score is computed as the sum of the score obtained from the root filter and sum of parts of the maximum (over the placement of that part) of the part filter scores minus a deformation cost associated with the deviation of the parts from their ideal location. A set of root filters and part filters are said to constitute a component of an object. Each object is modeled as a mixture of such components. The final score is then obtained as the maximum of the scores corresponding to each component. The mixture of components captures the variation associated with different view and orientations of an object. An illustration of the different filters associated with an object model is shown in Fig. 2.

The training of the model uses a technique known as the latent SVM (LSVM). The LSVM formulation is shown in Eq. 1, where each example x is scored by the function $f_w(x)$.

$$f_w(x) = max_z \left(w \cdot \phi(x, z)\right) \quad (1)$$

Here, w corresponds to a concatenation of the weights

associated with different filters. z represents the latent variables of the model, which indicate the configuration of the object in an image and the component label corresponding to it. $\phi(x, z)$ is the feature vector, which is the concatenation of the window of the feature map corresponding to the root filter and the sub-windows corresponding to the different parts. In general, the problem in Eq. 1 is convex in w for negative examples, but not for positive examples. The problem is made convex by fixing one possible latent variable for each positive example.

2.2. Optical Flow

The optical flow algorithm attempts to determine the displacement of pixels between two successive frames in a video based on assumptions of spatial coherence and small displacement. Let us represent the pixel at time $t+\tau$ located at the position vector $\mathbf{x} = (x, y)$ by $I(\mathbf{x}, t + \tau)$. The pixel position at time t can be obtained by considering a small displacement in time τ, denoted by the displacement vector $\delta\mathbf{x}, t, \tau$ as shown in Eq. 2.

$$I(\mathbf{x}, t + \tau) = I(\mathbf{x} - \delta(\mathbf{x}, t, \tau), t) \tag{2}$$

A Taylor series expansion of Eq. 2, neglecting the higher order terms gives us a constraint as shown in Eq. 3, where $\mathbf{V_x}$ represents the velocity of the pixel at position $\mathbf{x}$.

$$\nabla_{\mathbf{x}} I \cdot \mathbf{V_x} = -\frac{\partial I}{\partial t} \tag{3}$$

The equation has two unknowns in the components of the velocity vector $\mathbf{V_x}$, and is solved by imposing an additional condition based on the assumption of spatial coherence. In this project, we have made use of the Lucas-Kanade iterative implementation of the optical flow algorithm which assumes that the movement of pixels within a small neighborhood of the pixel under consideration between successive frames is small to slove Eq. 3.

2.3. Objectbank

[7] provided a method to represent images as a set of response-maps of object detectors, for high level visual tasks like image understanding. Pre-trained generic object detectors were used to build a feature, denoting the presence of objects at different scales and quadrants of an image. This representation was shown to capture semantic and spatial information of objects present in an image. They had further analyzed the choice of objects needed for a specific visual task. We extend on this idea to build a simple feature vector to represent a video as a collection of the best object sequences present in it. The idea of object sequences in explained in Sec. 4.2.

3. Approach

Given an object detector trained on images from a generic database (which will be refered to as the generic object model), we wish to gradually remove the dataset bias from the model and move towards a model (referred to as the dataset specific object model) more specific to the video dataset in hand. We first initialize the training procedure with a model trained on an image database. Secondly, we use this model in combination with an optical flow based filtering method to detect objects with high confidence and annotate corresponding frames from training videos. A flow diagram depicting the extraction of object sequences from one video is shown in Fig. 3. Thirdly, the newly annotated video frames along with the original database images are used to train a new object detection model. This procedure is repeated iteratively to train object detection models. The different steps are explained below.

We also discuss, the use of a small set of these object detectors to build a simple feature vector for event classification. These features will later be used in Sec. 4.2 to show the improvement in performance obtained by switching from the generic object model to the specific model.

3.1. Initialization

The event specific object is decided based on the evidential description of the event provided in the event-kit. In this project, we consider only one event related object for each event class. For instance, "skateboards" are chosen as the object relevant to the event class "attempting a board trick". The corresponding annotated images from the Imagenet database are used to train an initial part based model as described in [3].

3.2. Bounding box detection in video

This part based model is used to detect object bounding boxes from all training video frames. The top four bounding boxes B_i^j where, $j \in \{1, 2, 3, 4\}$ with highest detection scores in each frame is retained along with their corresponding scores S_i^j. These four bounding boxes are used to assign score values to pixels in the image to form a score-map. Each pixel in an frame from a video is assigned the score of the best bounding box it falls into. If a pixel does not fall into any bounding box, it is assigned a very high negative score S_{min}. Let $S_i(x)$ denote the score assigned to the pixel at position x in the i^{th} frame of a video. The score values are now filtered using a optical flow based filter to ensure that only objects which are consistently detected in a sequence of frames are retained.

3.3. Optical flow based temporal filtering

Optical flow can be used to track a dense set of points across a sequence of frames. For every pixel in a frame, the

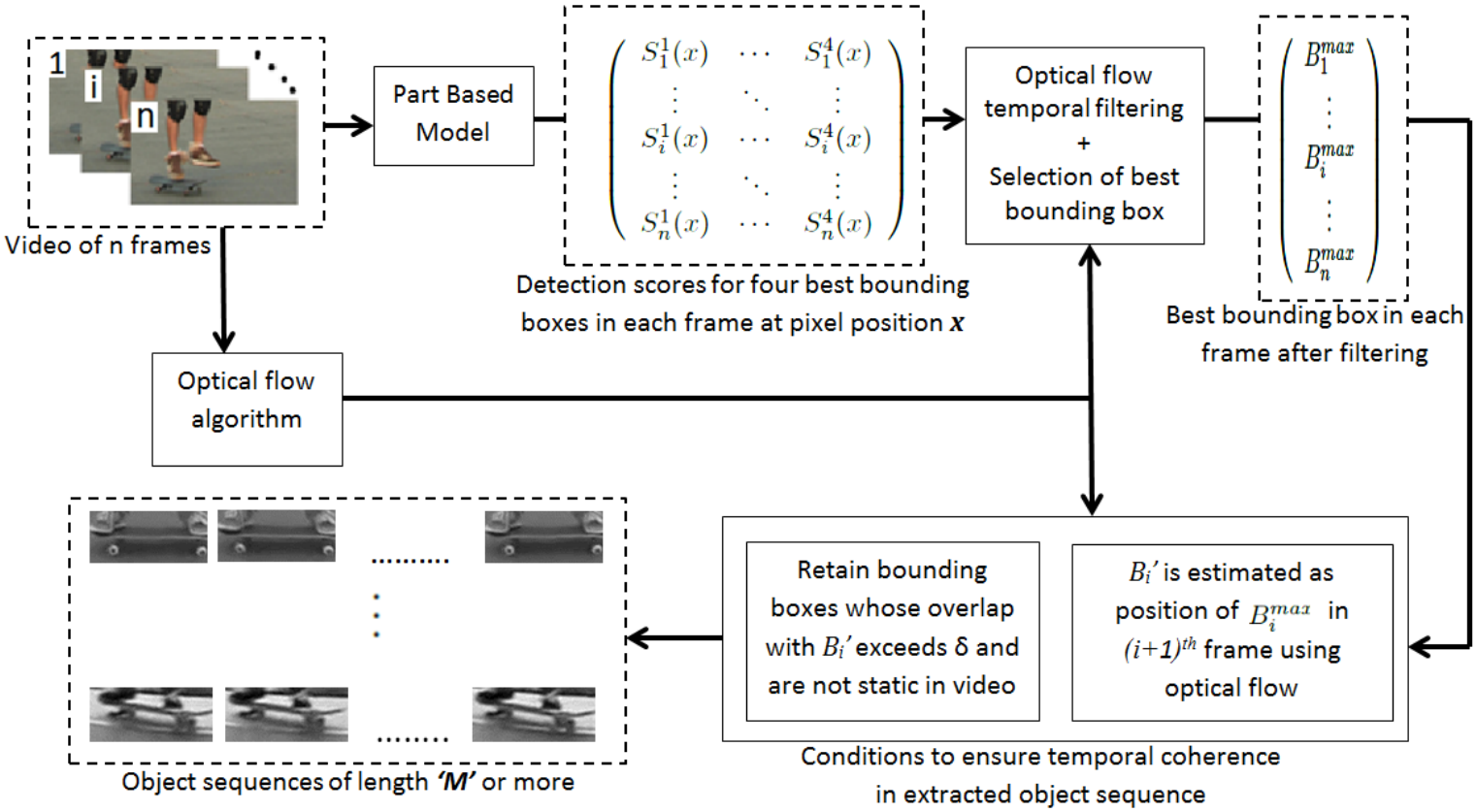


Figure 3. A flow diagram showing the extraction of object sequences from a video using the scheme expalined from Sec. 3.1 to Sec. 3.3

corresponding position in another frame can be obtained. Let $u_{i,k}(x)$ denote the displacement of the pixel at position x from the i^{th} frame to k^{th} frame. We use this information to filter the scores S_i. This filtering is carried out across a window of frames. Let $(2N+1)$ denote the window size and $R_i(x)$, the modified score at pixel x of i^{th} frame. Then,

$$R_i(x) = \sum_{k=-N}^{N} w_k S_{i-k}(x + u_{i,i-k}(x)) \tag{4}$$

The filter coefficients w_k are chosen according to a Gaussian kernel. This smoothens out any irregularities in object detection across successive frames. a toy example is demonstrated in fig .4, for a window size of 3. After obtaining R_i, the filtered scores R_i^j of bounding boxes B_i^j in images are computed as the average of all pixels belonging to the bounding box.

$$R_i^j = \frac{\sum_{x \in B_i^j} R_i(x)}{|B_i^j|} \tag{5}$$

Here, $|.|$ represents the size of the bounding box. Having obtained the filtered score values, we still need to eliminate a large number of spurious detections and retain only

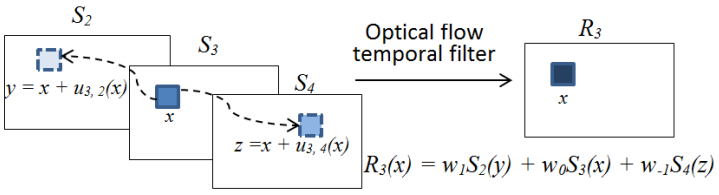


Figure 4. A toy example showing the temporal filtering carried out at a pixel x in the 3^{rd} frame of a video. We have considered a temporal window of size 3. y and z are the positions of the pixel in the 2^{nd} and 4^{th} frame respectively, estimated by optical flow. S_i represents the original score-map obtained from the detection algorithm. R_3 is the final score-map after filtering. w_i represents the filter coefficients.

"good" detections. Again, we impose the criteria that good detections will be consistent across a sequence of frames. Hence, we retain the detection only if the same region is detected in neighboring frames as well. We impose a set of hard conditions to achieve this. Let, B_i^{max} be the box with the highest score R_i^{max} in the i^{th} frame. We reject all other bounding boxes in the frame. Let B_i' denote the bounding box obtained by displacing B_i^{max} from i^{th} frame to $(i+1)^{th}$

using optical flow. We reject the detection B_i^{max} if the overlap between the dispalced box B_i' and B_{i+1}^{max} is less than a threshold δ. Then, we move a window of size M in the temporal domain and retain only those detection sequences which have a length greater than M. A detection sequence in this context refers to a set of consecutive frames where an object has been detected (bounding box retained according to the previous conditions). Finally, the average velocity of the object in the sequence is computed using optical flow measurements. The detection sequence is rejected if this velocity is less than a threshold τ. This condition helps eliminate noisy detections particularly from background clutter. Moreover, objects which are static through a sequence of frames will add less value to training. The conditions are enumerated below.

1. Only the bounding box B_i^{max} with naximum score in each frame is retained
2. B_i^{max} is rejected, if the overlap between the dispalced box B_i' and B_{i+1}^{max} is less than δ
3. Only detection sequences(consecutive frames, where a bounding box has been retained) with length greater than M are retained
4. A detection sequence is rejected if the average velocity of the object in the sequence is less than τ

These stringent conditions enforce the criteria, that only good detections are retained. This method is used to extract object sequences from all training videos belonging to the event class.

3.4. Iterative training

Let $P_{Imagenet}$ represent the object detection model obtained by training only with the Imagenet object images. The object detection results from Sec. 3.3 are used to obtain a set of image annotated with the bounding box information. These new images are now added to the pool of Imagenet images to train a new object detection model $P_{Imagenet+video}$. Alongside, another object detection model P_{video} is obtained by training only with the objects detected from Sec. 3.3. Both the models are cross validated on image frames from test videoset using $5 - fold$ validation. The model with the better average precision score is used in the next iteration. The steps discussed in Sec. 3.2 and 3.3 are repeated with this new object detection model. Finally an object detection model trained only on frames from training video sequence is obtained which outperforms the original model $P_{Imagenet}$. It is to be noted that in our experiments the negative training examples remain consistent throughout all iterations. However, this need not be the case. An equal number of negative training samples can also be extracted in a similar fashion form training videos and used for training.

3.5. Choice of objects

Experiments in [7] have shown that, rich semantic information in image scenes can be captured with only 20 object filters. By object filters, we refer to a convolution filter which would provide a response-map denoting the presence of a specific object at different image locations. It was further hypothesized and verified that the distribution of objects in images followed Zipf's law. Hence extending this idea, with a choice of few object detectors, we should be able to extract meaningful object related information from videos. The choice of these objects is naturally dependent on the evidential description of events provided in the event data kit. The evidential description provides a list of attributes related with each event. The objects mentioned in this description can be used for iterative training of object model and subsequent event identification. If more than one event class share the same attribute (the object "person" for instance is associated with a large number of event classes), the iterative training can be carried out by extracting video frames from videos of all event classes which share this attribute. Due to computational and time limitations, we restrict the analysis to two event classes and two objects in our experiments. "skateboards" and "tires" are the prominent objects in the event classes "Attempting a board trick" and "Changing a tire" respectively. The presence and absence of these objects would almost provide sufficient information to distinguish these two classes. Hence, experiments in Sec. 4.2 will use these object classes only. It is to be noted that a more rigorous treatment should involve, considering all objects mentioned in the evidential description and finally selecting a smaller set based on a mutual information or similar criterion often used in feature selection. For the purpose of this report, we have bypassed this step and assumed that the two object classes "skateboard" and "tire" would provide maximum information about the event classes considered.

3.6. Features for video classifcation

In order to gain some insight into the actual utility of the scheme, a toy experiment is demonstrated in Sec. 4.2, where videos are classified into two categories based on a set of features. In this section, we describe a simple feature for videos similar to the Objectbank scheme for scene classification. Until now, we have looked at an iterative training method for developing effective object detectors for video detection. Once the object detectors are available, a video can be described in terms of the response of video frames to the object detectors. More specifically, we are interestd in the best sequence corresponding to an object in an image. We extract object sequences (of a fixed length M) with the same criteria mentioned in Sec. 3.3. However, an object sequence may not be present in a video always satisying these

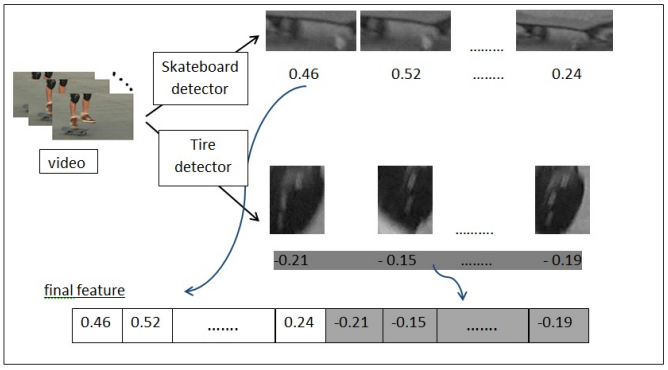
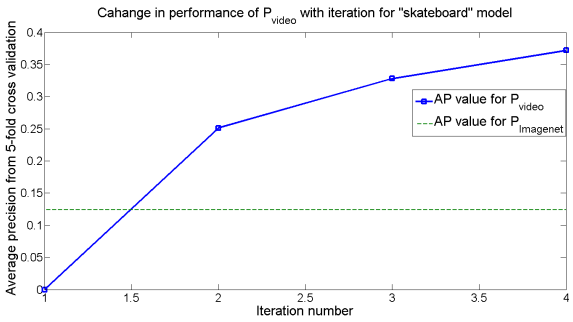


Figure 5. An illustration of the feature construction from a video using two object detectors. The top row shows the best "skateboard" sequence extracted from the video along with detection score for each frame. The bottom row shows the best "tire" sequence and corresponding scores below it. The two score vectors are concatenated to form the feature vector.

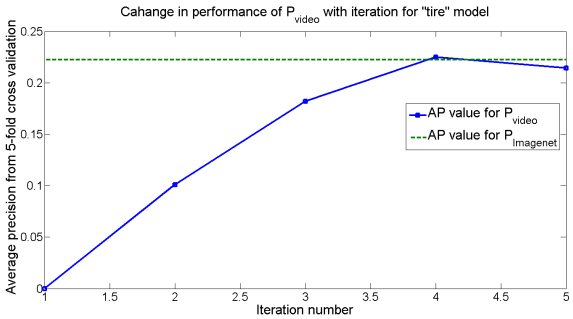
stringent conditions. Hence, we gradually relax the overlap threshold δ, till we obtain a "good" sequence. In the presence of multiple object sequences, we retain the sequence with the highest object detection score averaged across all frames of the sequence. This is similar to the Objectbank approach for scenes, where the highest responsemap score in spatial bins is concatenated to represent an image. Once the object sequence is obtained, we build a feature vector by concatenating the detection scores in each frame to form a feature vector of length M. Hence, by running n object detectors (pertaining to n objects like skateboard, tire, $\cdots$) we can obtain a feature of length nM by concatenating all these scores together. An example is demonstrated in Fig. 5. It is to be noted that, although we restrict ourselves to one sequence, a more rich feature can be obtained by considering multiple object sequences in an object and also taking into account the trajectory of such objects in the sequence.

4. Experiments

The experiments presented in this section were carried out on two event classes "Attempting a board trick" and "Changing a vehicle tire" from the Trecvid [9] event kit. Each event class contains $100+$ videos with variable length (usually around a 1000 frames). We have used two component models with six parts each for object detection. As mentioned in Sec. 3.5, we present results by considering detectors corresponding to two object classes "skateboard" and "tires". We first demonstrate some experiments to justify the use of iterative training, followed by a simple binary event classification exercise to show the gain of the iterative training scheme in the context of event detection.



(a)



(b)

Figure 6. Change in performance of the P_{video} with time is shown for (a) skateboard and (b) tire models. The dashed line corresponds to the average precision of $P_{Imagenet}$.

4.1. Dataset specific object detector

In this section we compare the performance of the generic object detector and the dataset specific object detector developed as discussed in Sec. 3. We build two object detectors ("skateboard" and "tire"), each corresponding to one event class. For each of the object models, training was iteratively carried out by extracting objects from a training set of 40 videos from the corresponding event class. The resultant models $P_{Imagenet+video}$ and P_{video} are evaluated on a set of $170+$ manually annotated images obtained from a test video set of 40 videos belonging to the corresponding event class. $5-$ fold cross validation is carried out on these images to obtain an Average Precision (AP) value for each model at the end of each iteration. During each iteration of the $5-$ fold validation experiment, the model is tested on 4 folds to identify the best detection threshold for the part based model. This threshold is then used to evaluate the AP value on the remaining 1 fold according to the PASCAL voc benchmark. The AP values reported are the average of the 5 AP values obtained from the 5 iterations of 5-fold validation. The progress of the iterative training is plotted in Fig. 6 for P_{video} of the two objects. The performance of all the three models after 3 iiterations is shown in Fig. 7. This variation in performance for the two models are discussed below

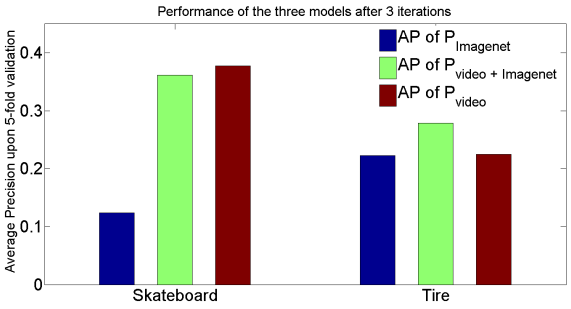


Figure 7. The Average Precision value upon 5-fold validation for the three models $P_{Imagenet}$, $P_{Imagenet+video}$ and P_{video} are shown for the tire and skateboard objects.

4.2. Event classification

In this experiment, we build feature vectors from videos as described in Sec. 3.6 and explore the classification results under different conditions. We run experiments on a set of 50 videos, 25 belonging to the event class "Attempting a board trick" and 25 belonging to the class "Changing a vehicle tire". The results are presented in the form of average Precision-Recall curves by averaging the curves over all 50 videos. For a given video, the remaining videos in the test set are ranked according to the Euclidean distance of their feature from the test video feature under consideration.

First, we build a feature vector of size 10 by extracting the best 10-frame skateboard sequence from a video and concatenating the detection scores. In this experiment, we have made use of only the "skateboard" model. The PR curve is compared for P_{video} and $P_{Imagenet}$ models in Fig. 4.2. Similarly, the results obtained by using feature vectors of size 10 from "tire" models (by extracting the best "tire" sequence from videos) are plotted in Fig. 4.2. Next, results are obtained by concatenating both the features from "skateboard" and "tire" models as shown in Fig. 5. The results using this feature obtained from P_{video} and $P_{Imagenet}$ are shown in Fig. 4.2. It is seen that the average precision value is greater for P_{video} in all three cases. The model obtained from iterative training is seen to perform better.

Finally, we explore the information addition by considering more than one object models for P_{video}. The PR curves are shown in Fig. 4.2. The Mean Average Precision values for each case is also shown in the figure. As expected, the performance improves by extracting more object related information from videos. This result can also be seen in the light of results presented in [7], where scene classification results improve with addition of more object filters.

5. Discussions

In this section, we discuss the possible reasons for the observed change in model performance with introduction of iterative training. We consider the change in AP values in the 5-fold validation experiment.

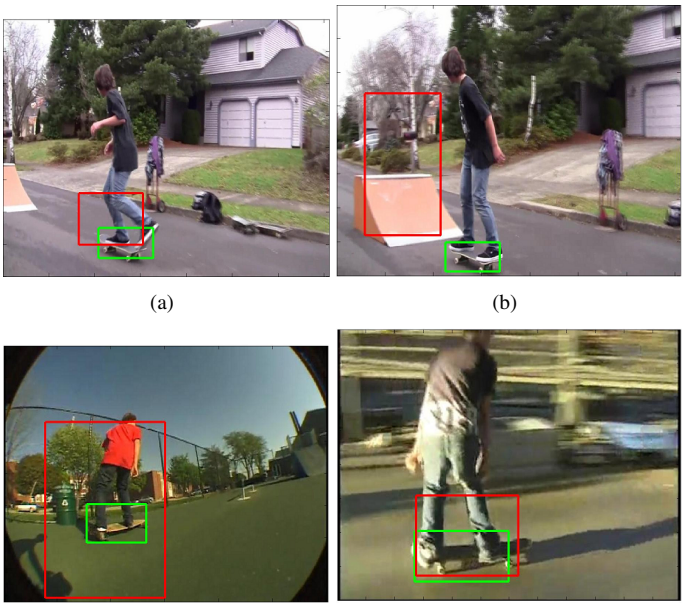


Figure 9. Three sample video frames from test videos is shown, where P_{video} performs better. The green bounding box orresponds to P_{video}, while the red one corresponds to $P_{Imagenet}$. Only the best detection in a given frame is shown in this example.

5.1. Skateboard model

After the third iteration, the P_{video} outperforms the remaining models. In other words, we have gradually moved from an object detector trained on a generic image database to a detector specific to the video dataset of interest. The number of training objects detected at the end of each iteration also increases with the number of iterations (from 180 after initialization to 640 after 2^{nd} iteration). It was seen that, roughly 85% of the detected objects pertained to a skateboard or atleast a large part of the skateboard, while the remaining were spurious detections. The performance of $P_{Imagenet+video}$ and P_{video} is also seen to be vastly better than $P_{Imagenet}$. Some sample video frames are also shown in Fig. 9, where P_{video} is seen to perform better than $P_{Imagenet}$.

The improvement in performance can be accounted to a favourable change in the model components as well as some contextual details added by the iterative training. It was observed that the Imagenet skateboard images contained a large number of images of independent skateboards, sometimes in vertical orientation. However, the skateboard images extracted from the videos correspond to people skating on the skateboard. Hence, the segmented skateboard images contain a part of the human feet segmented as well. Also,

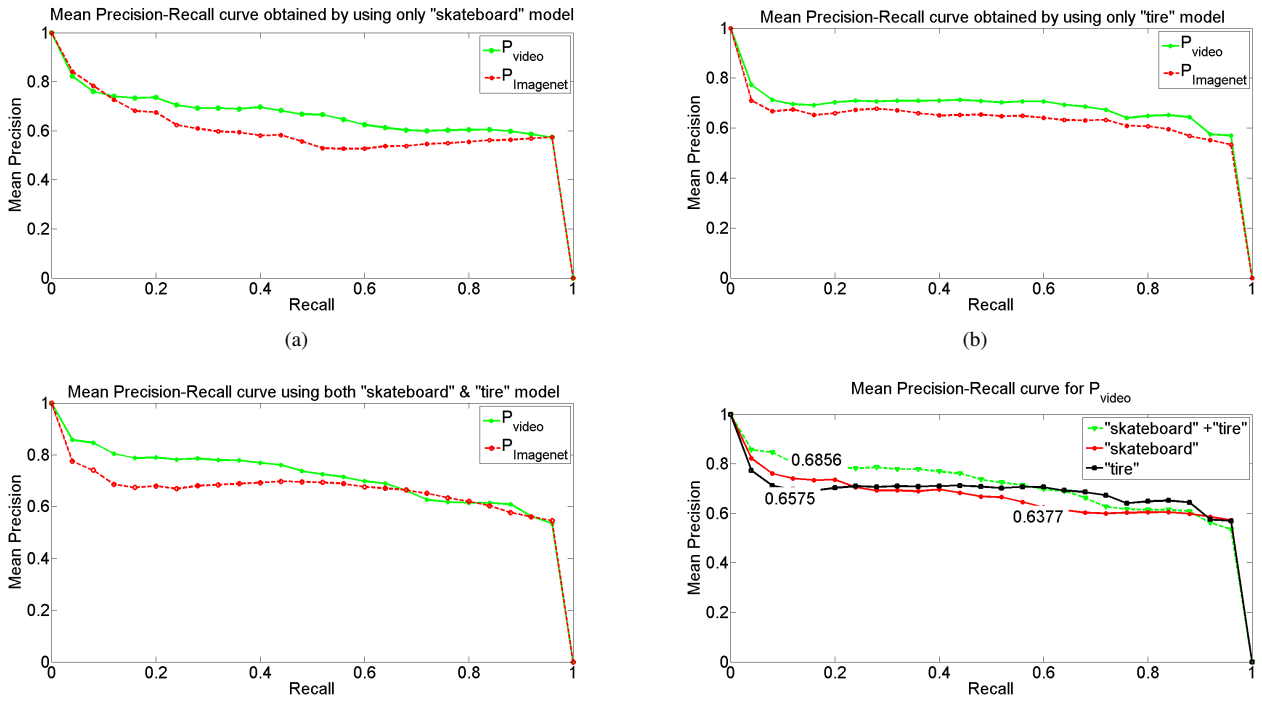


Figure 8. The PR curves are plotted for P_{video} (green) and $P_{Imagenet}$ (red) using (a) feature of size 10 built from only "skateboard" model, (b) feature of size 10 built from only "tire" model and (c) feature of size 20 built from both "skateboard" and "tire" model. The PR curves are plotted for P_{video} using features from "skateboard" only (red curve, MAP = 0.6377), "tire" only (black curve, MAP = 0.6575) and both the models(green curve, MAP = 0.6856)

the orientation of the skateboard was always horizontal and pertained to the longitudinal view of the skateboard. This led to a change in the components of the P_{video} model as compared to $P_{Imagenet}$. It can be seen from Fig. 10, that both the components of P_{video} correspond to the horizontal longitudinal view, unlike the $P_{Imagenet}$ model.

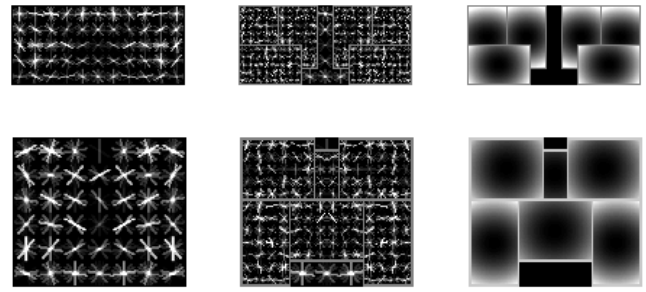
5.2. Tire model

The performance of the P_{video} model corresponding to the "tire" model does not show significant improvement as compared to $P_{Imagenet}$. It was observed that the model components did not change significantly with iterative training. The extracted images from the videos were also found to be close to the images from Imagenet. This result is expected, since "tire" is a low level object, in the sense that it has smaller variation with changing orientation and lesser parts. Hence, no additional information is added to the model by iterative training.

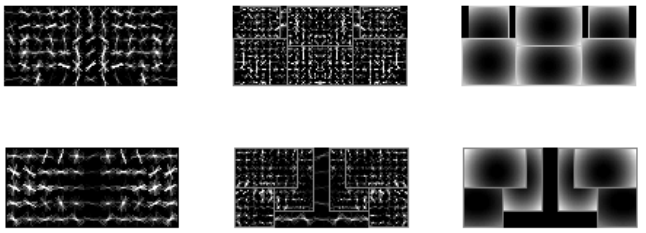
6. Limitations

A major limitation of the proposed method is the excessive computation time for iterative training. The part based model requires $\approx 2secs$ for bounding box detection from one image. The iterative training runs the detectors on 40 videos with more than 1000 frames for each iteration. Alongside, the optical flow algorithm requires $\approx 2secs$ per frame. Hence, each iteration along with the time required for training the model requires more than 24 hours. Hence, a large amount of time is spent in training a single object model iteratively. Similarly, extracting the best object sequence from a video is also time consuming. This is not acceptable and a more practical approximation of the scheme should be used. Some possible solutions are to use a model without parts and lesser components. A less dense feature tracker like KLT tracker can be used. Pre-computation of these tracks can tremendously reduce the computation time.

Although, we have used models to extract object sequences from event videos, some videos might not be completely distinguishable in terms of objects. For example, the events "Feeding an animal" and "Grooming an animal" would share similar object attributes. The main sifference lies in the interaction between objects in such cases. However, object sequences in videos do provide valuable information. The trajectories of these objects in the videos should be used to build richer features.



(a)



(b)

Figure 10. The model visualization for the skateboard models corresponding to (a) $P_{Imagnet}$ and (b) P_{video} after the final iteration. It is seen that both the components of the P_{video} correspond to a longitudinal view of the skateboard unlike $P_{Imagenet}$.

7. Conclusion

In this project, we proposed a method to build a dataset specific object detector from a generic object detector by iteratively extracting object instances from the training videos and training newer models. While extracting object sequences from videos, we used the assumption of temporal coherence among video frames to segment objects with high confidence. We tested the scheme on two objects, "skateboards" and "tires" by training on two event classes. It was observed that the "skateboard" model underwent a significant change on iterative training, as only context (like longitudinal orientation and presence of human feet on skateboard) relevant to the video dataset was retained, making it perform better on the test videos. The newly trained models were used to extract a simple feature for event classification. The feature denoted the presence of an object sequence in a video. It was again shown that the iteratively trained model performed better than the model trained on a generic image database. However, the iterative training process is currently very computation intensive, and a faster implementation needs to be developed with perhaps fewer components in the model. The next step in the project would be to explore the effect of introducing more object models for event classification and building a richer feature to encompass the interaction between different objects in a video. The object related information could also be combined with lower level features for better performance.

References

[1] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[2] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2011 (VOC2011) Results. http://www.pascal-network.org/challenges/VOC/voc2011/workshop/index.html.

[3] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9), 2010.

[4] P. F. Felzenszwalb, R. B. Girshick, and D. McAllester. Discriminatively trained deformable part models, release 4. http://people.cs.uchicago.edu/ pff/latent-release4/.

[5] B. K. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 17(1-3):185 – 203, 1981.

[6] Y. Ke, R. Sukthankar, and M. Hebert. Event detection in crowded videos. In *ICCV 2007*, pages 1–8, 2007.

[7] L.-J. Li*, H. Su*, E. P. Xing, and F.-F. Li. Object bank: A high-level image representation for scene classification and semantic feature sparsification. In *Proceedings of the Neural Information Processing Systems (NIPS)*, 2010.

[8] D. Ramanan and D. A. Forsyth. Automatic annotation of everyday movements. In *NIPS*, 2003.

[9] A. F. Smeaton, P. Over, and W. Kraaij. Evaluation campaigns and trecvid. In *MIR '06: Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval*, pages 321–330, New York, NY, USA, 2006. ACM Press.

[10] A. Torralba and A. Efros. Unbiased look at dataset bias. In *CVPR11*, 2011.