SCC: Semantic Context Cascade for Efficient Action Detection

Fabian Caba Heilbron, Wayner Barrios, Victor Escorcia and Bernard Ghanem
King Abdullah University of Science and Technology (KAUST), Thuwal, Saudi Arabia
http://www.cabaf.net/scc

Abstract

Despite the recent advances in large-scale video analysis, action detection remains as one of the most challenging unsolved problems in computer vision. This snag is in part due to the large volume of data that needs to be analyzed to detect actions in videos. Existing approaches have mitigated the computational cost, but still, these methods lack rich high-level semantics that helps them to localize the actions quickly. In this paper, we introduce a Semantic Cascade Context (SCC) model that aims to detect action in long video sequences. By embracing semantic priors associated with human activities, SCC produces high-quality class-specific action proposals and prune unrelated activities in a cascade fashion. Experimental results in ActivityNet unveils that SCC achieves state-of-the-art performance for action detection while operating at real time.

1. Introduction

Imagine you would like to find and share videos in your digital archives about the remarkable moments you had playing beach volleyball in Hawaii (refer to Figure 1). To do this, you have to scan every video and determine whether or not the moments you are looking for are present in each video. To optimize your search time, you would probably scroll through the archives quickly and stop to check time instances, where you saw a *beach*, *volleyball net*, or *volleyball*. At those times, you would scroll or play the video slower to determine whether this part of the video is one of the special moments you are looking for. If it is not, you resume the coarse-to-fine temporal search through the videos, until you have exhausted them all.

This particular search problem not only afflicts people looking for memorable moments, but it also hinders various real-world tasks ranging from consumer video summarization to surveillance, crowd monitoring, and elderly care. There is obviously a need for efficient and accurate automated methods that can search and retrieves events and activities in video collections, formally known in the vision community as action/activity detection in long untrimmed videos. Despite the great research efforts that have been made on the topic of action recognition and detection, the goal of accurate and fast detection remains elusive in our automated visual systems.

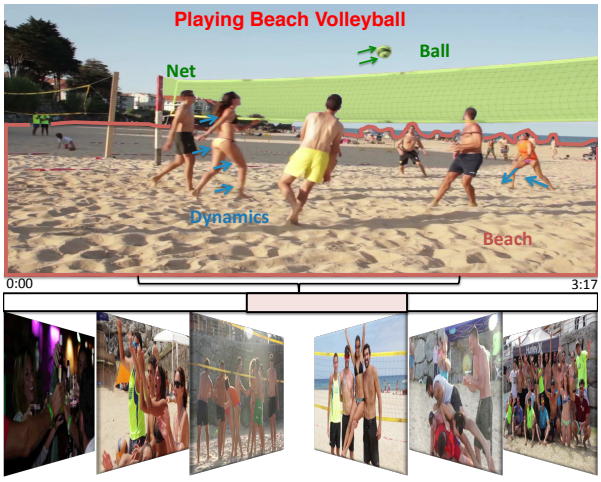


Figure 1. *Playing beach volleyball* is more than the sum of people running, jumping, and hitting a ball. It inherently implies an outdoors beach, a volleyball net, a volleyball and humans interacting in a particular way. Our approach leverages this rich and discriminative semantic information (namely objects and places) to determine when activities of interest occur in long, untrimmed videos in an efficient and effective way.

First attempts in action detection apply activity classifiers exhaustively over the video at each time location and at multiple temporal scales [8, 10, 32]. Despite the good detection performance they achieve in small-scale and controlled scenarios, this computationally expensive approach is infeasible for large-scale video analysis applications. To overcome the computational demand of these traditional methods and inspired by progress in the object detection domain [18, 35, 36], recent approaches [5, 9, 37] develop methods that quickly scan a video to generate temporal segments, where general activities are likely to exist. In doing so, activity classifiers are only applied to few candidate segments, thus, significantly reducing the computational overhead. However, these detection approaches ignore semantic context priors (*e.g.* objects and scenes) in localizing actions, even though they have been shown to be quite effective in

describing actions and boosting action classification performance [19, 21, 29, 50]. In this paper, we embed the use of semantic context in the detection action process.

Consider again the video sequence in Figure 1.The existence of the volleyball, net, and beach in a video frame is a good semantic prior that lends visual evidence encouraging a detection of the *playing beach volleyball* action to include this frame. In other words, we argue that **(i)** semantic context in the form of action-object and action-scene relationships (*e.g*. co-occurrence) can help guide the temporal localization of actions in an untrimmed video. Moreover, the lack of this context can also be informative. For example, knowing that video frames do not contain a *dog* and are taken *indoors* discourage the detection of the *walking the dog* and *shoveling snow* actions. In addition to improving localization, we also argue that **(ii)** action-object and action-scene relationships can be exploited to quickly prune out or disregard actions that are unlikely to exist in a video segment *without* applying an expensive action classifier. This cascaded approach is especially useful when the number of action classes is large, as is the case in many activity datasets nowadays (*e.g*. ActivityNet [4]). In fact, we realize that claims **(i)-(ii)** are validated by observing how humans scroll through long videos, while searching for particular action classes (more details in Section 3).

Contributions. The core idea of the paper is to introduce a model that embraces rich semantic context information that has strong associations with human activities. Specifically, the contributions are twofold. **(1)** We introduce a new *Semantic Context Cascade* (SCC) model, which exploits action-object and action-scene relationships to improve the localization quality (*i.e*. recall) of a fast generic action proposal method and to quickly prune out unrelated actions in a cascade fashion. These two features lead to an efficient and accurate cascade pipeline for detection. **(2)** When applied to ActivityNet [4], the most diverse large-scale dataset for activity detection, our SCC model achieves state-of-the-art performance, while significantly reducing computational cost, as compared to state-of-the-art detectors.

2. Related Work

Modeling context for actions. Several approaches have incorporated context cues to boost action recognition performance in controlled scenarios [14, 16, 25, 26, 34, 47, 49]. Marszaek *et al*. [29] show the relevance of the co-occurrence between actions and scenes to design useful visual representations for retrieving short actions in movie clips. In a similar spirit of empowering visual features for action understanding, [14, 19, 20, 25, 34, 47] show that implicit and explicit modeling of the relationships between objects in the video allows to discriminate action occurring in videos, especially by reducing the confusion between actions with similar motions such as *drinking* and *smoking*. More recently, Jain *et al*. [21] extend this idea further by conducting a large-scale study that reveals a strong co-occurrence between actions and a sparse number of objects. In the same spirit as this work, Wu *et al*. [50] use high capacity neural networks to learn object, scene, and action relationships with the end goal of improving activity classification. Although the efficacy of context cues has been successfully proven to help action classifiers be more discriminative, previous approaches do not explore these ideas to tackle the challenges in action detection. To the best of our knowledge, our work is the first to address action detection by exploiting semantic information garnered from action-object and action-scene relationships at large scale.

Action detection. Following the success and ubiquity of 2D object proposals in 2D object detectors [18], spatio-temporal proposals [43, 48, 53] and temporal activity proposals [5, 9, 30, 37] have emerged as a key pre-processing step to avoid the exhaustive sliding window approach for action detection [8, 10, 32]. Closely to our work, Caba Heilbron *et al*. [5] introduce a sparse learning framework to efficiently scan videos (10FPS) and produce a set of high fidelity temporal proposals that are likely to contain actions. The contemporary work of Shou *et al*. [37] provides a temporal proposal module that helps a multi-stage system to filter out background segments (60FPS). To further improve upon the quality and computational efficiency of prior work, Escorcia *et al*. [9] speed up the proposal generation step even further by employing deep learning models and memory cells (130FPS). Another line of research has explored the use of attention models to focus in temporal snippets of a video [31, 52]. However, both types of approaches (*i.e*. action proposals and attention models) lack an explicit encoding and use of action associated semantic information (*e.g*. action relationships with objects and scenes), which we argue is important in quickly detecting human activities.

3. Motivation

There is clear evidence that humans use context information and semantic priors to successfully perform visual tasks [3, 15]. To validate this argument in the realm of action detection and to motivate our use of semantic context in a cascade for improving action localization, we conduct an online user study, where human subjects are given a clear and concise task, namely to annotate the start and ending times of particular activities in video sequences. To identify temporal parts of the video where the user focuses, we log all his/her interactions with the user interface (video browser). Possible actions on the video level are include sliding the time bar left or right to quickly search for the start and ending time respectively, as well as, jumping directly to any temporal point in the video.

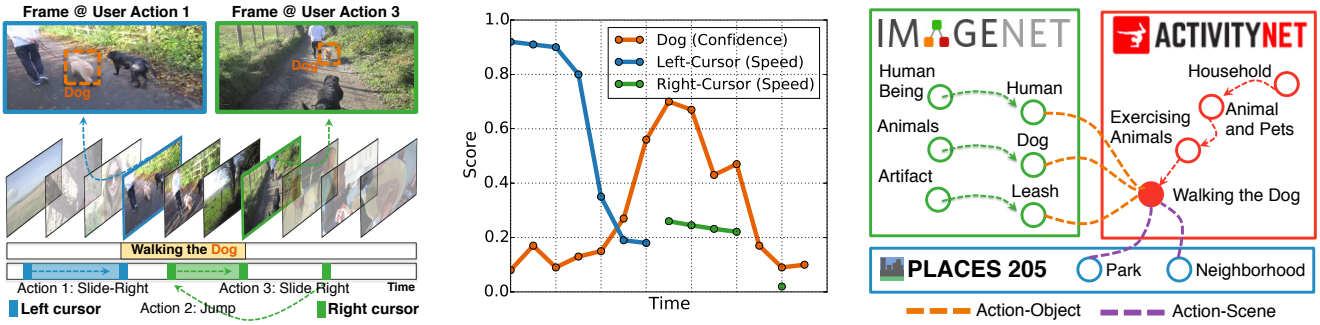


Figure 2. The **left** illustration depicts the sequence of actions that a human follows to annotate the temporal boundaries of walking the dog in a video. **Center** Speed of left and right cursors against object detector response (dog). **Right** Example of action-object and action-scene links. We observe a strong correlation between semantic changes (*e.g.* after dog appears in the video) and temporal points of the video where the user focus. To benefit from the rich semantic information that action-object and action-scene relationships provide, we conduct an annotation effort that links three large-scale datasets.

Our study reveals that there is a strong correlation between semantic changes and the temporal parts of the video, where the user focuses. These temporal parts tend to be strongly related with semantic priors of the intended activity. Consider Figure 2 (left), which illustrates an example of the sequences of steps executed to annotate the action *walking the dog*. As in this example, our study finds that users tend to scan the video quickly until a semantic prior associated with the action appears in the video, in this case, the dog. This observation motivates our semantic context cascade (SCC) model, but also motivates us to annotate semantic relationships between actions, objects, and scenes.

Figure 2 (center) shows a typical example of how users annotate the activity *walking the dog*, *i.e.* how they move the cursors, accompanied with the temporal evolution of a *dog* detector response. Interestingly, peaks in the detector responses (changes in object appearance) correlate with minimum cursor speed. This behavior indicates that semantic context associated with the activity class (*i.e.* presence of dog) is used by the subject to quickly reach a part of the video that aligns with this context. Then, at a much slower pace, the user makes use of the semantic information along with an understanding of the activity class to define the bounds of this activity instance in the video. Details of this study and more user examples can be found in the supplementary material.

To benefit from action-object and action-scene associations, we first need to infer such relationships first. To do this, we conduct an annotation effort that links three of the largest datasets in computer vision: ImageNet [7], ActivityNet [4], and Places205 [55]. Given that our target is action detection, we annotate the set of objects and scenes that are relevant for each category in ActivityNet. We rely on Amazon Mechanical Turk workers to obtain multiple textual descriptions for each category in ActivityNet [4]. Then, we post-process this information to get candidate objects and scenes that are potentially relevant to the activities. We manually define the semantic relationships (action-object and action-scene), using the existing categories in ImageNet[7] and Places205[55]. Figure 2 (right) shows an example of the annotated links for the activity *walking the dog*. Later, we will use the links between the hierarchical organizations of activities, objects, and scenes to improve the localization quality of extracted action proposals and prune out action classes that are improbable in these proposals. The mined relationships from this annotation effort and further details about the annotation protocol can be found in the supplementary material.

4. Semantic Context Cascade (SCC) Model

Our goal is to develop a model that detects when and which actions (among a large set) happen in a video. Our primary challenge is to design this model in such a way that it produces a reliable detection while keeping the computational footprint low, so it can be feasible at large-scales (*i.e.* a large number of long videos and a large number of action classes). Therefore, we propose our Semantic Context Cascade (see Figure 3) model that exploits the efficacy of high recall action proposals, the action discriminative cues in a video's semantic context (objects and scenes), and the power of action classifiers, to perform action detection in a cascade fashion. The cascade has three stages: (1) action proposals, (2) semantic encoder, and (3) action classifier. Each of these stages is intended to progressively prune candidate detections that have neither *actionness* nor relevant semantic information.

4.1. Stage 1: Action Proposals

Action proposal methods have proven their ability to quickly generate temporal segments at different scales with high recall within a video [5, 9, 37]. Given that speed and recall are crucial in our design, we choose DAPs [9] to extract action proposals from untrimmed videos. This approach allows us to efficiently scan the video at 130FPS and produce high fidelity action proposals at multiple scales in a single pass. For completeness, we give a brief descrip-

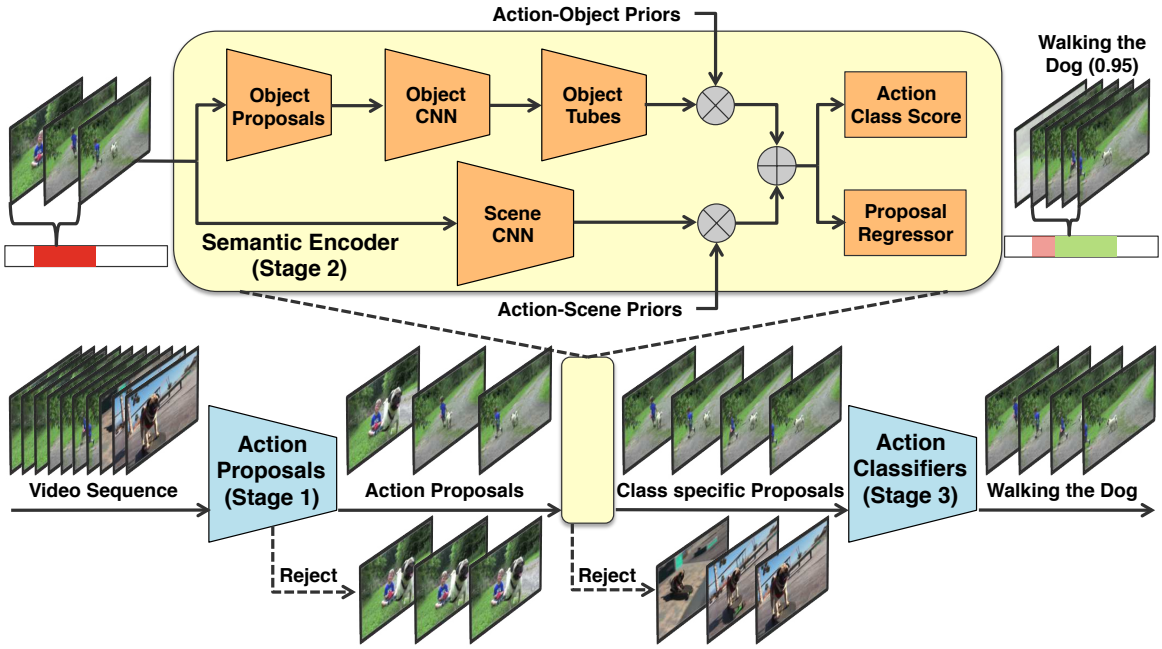


Figure 3. We propose a multi-stage cascade model to efficiently scan a video and determine when activities of interest occur. We rely on efficient *action proposals* to prune out segments where it is unlikely to find activities. Later, a *semantic encoder* combines the temporal information about objects and scenes along the segment with the prior knowledge about action-object and action-scene relationships to refine its time boundaries or prune it out in a class specific way. Finally, the last pool of segments are further analyzed by an *action classifier* which determines the probability that an adjusted segment belongs to a particular activity.

tion of the DAPs architecture, which contains four modules. The *visual encoder* represents the visual information in a video as activations from a pre-trained C3D [41] network. A *sequence encoder* (namely an LSTM) models the evolution of the C3D features over time for the purpose of generic action localization. Then, a *localization module* generates start and ending times for candidate proposals of different temporal lengths throughout the input video. Finally, the *prediction module* assigns a confidence score to each action proposal, based on the probability of it containing an activity of interest. As such, the output of Stage 1 for each video is a set of n_p temporal proposals, denoted by $\mathbf{P} = [\mathbf{p}_1|\cdots|\mathbf{p}_{n_p}]$ where $\mathbf{p}_i \in \mathbb{R}^2$ encodes the temporal location of the i^{th} proposal.

4.2. Stage 2: Semantic Encoder

Inspired by the correlation between object detector responses and the refinement of temporal activity annotations done by humans (refer to Section 3), our semantic encoder leverages the semantic context of each segment to improve its localization and action likelihood as Figure 3 depicts. Specifically, we exploit the prior knowledge coming from the link between objects and scenes associated with activities and the temporal activations of objects and scenes along the segment, to achieve this task in a class-specific manner. In this way, our semantic encoder improves the results of the first stages by: rejecting proposals that are not of interest, adjusting the start and end times of each proposal for better class-specific localization, and marginalizing the cost of computationally expensive action classifiers needed for the pool of proposals by pruning classes that are unlikely to exist in each action proposal.

Formalizing semantic context. We encode the annotated action-object and action-scene relationships as a binary matrix $\mathbf{L}_o \in \{0,1\}^{o\times c}$ and $\mathbf{L}_s \in \{0,1\}^{s\times c}$ respectively. Here, c denotes the number of action categories we are interested in, o the number of objects linked to the c actions, and s the number of linked scenes. In our experiments, o, s and c are 440, 48 and 200 respectively. For example, if action j is linked to object i and scene k, then $\mathbf{L}_o(i,j) = 1$ and $\mathbf{L}_s(k,j) = 1$; otherwise, they are 0.

Expected Stage 2 Output. In what follows, we will explore how the original proposals in $\mathbf{P}$ can be transformed and pruned into the following arguments, so they are later fed into the action classifier in Stage 3.

1. **Updated Proposal Locations**: A tensor $\mathbf{P}_{SCC} \in \mathbb{R}^{2\times m\times c}$ encodes the $m \leq n_p$ action-specific proposals left after filtering. Similar in spirit to the region proposal network (RPN) [35], the location of each filtered proposal is adjusted according to each action class.

2. **Class-Specific Action Scores**: A binary matrix $\mathbf{S}_{SCC} \in \{0,1\}^{c\times m}$ encodes which action classifiers per proposal need to be applied in the next stage. For example, if column i in $\mathbf{S}_{SCC}$ contains only one 1 at

row j (*i.e.* $\mathbf{S}_{SCC}(j,i) = 1$), then only the action j classifier is applied to adjusted proposal i in Stage 3.

Encoding action-object relationships.

Previous work has shown the importance of exploiting object representations for action classification [19, 21, 50]. However, these methods use this semantic information at the global video level for description only. Instead, we encode the spatiotemporal evolution of object detections in each action proposal. To do this, we first extract generic object proposals, namely EdgeBoxes [56], from frames within each action proposal at 3FPS. On each object proposal, we apply a ResNet [17] classifier finetuned on the o object classes in ImageNet that were found to be semantically relevant to the c action classes. Note that we could not finetune an end-to-end object detector (*e.g.* Faster R-CNN [35]) here, since no ground truth object detections are available for ActivityNet. Also, the set of o objects contains many more classes than those in available detectors trained on ImageNet [36], COCO [27], or other detection datasets.

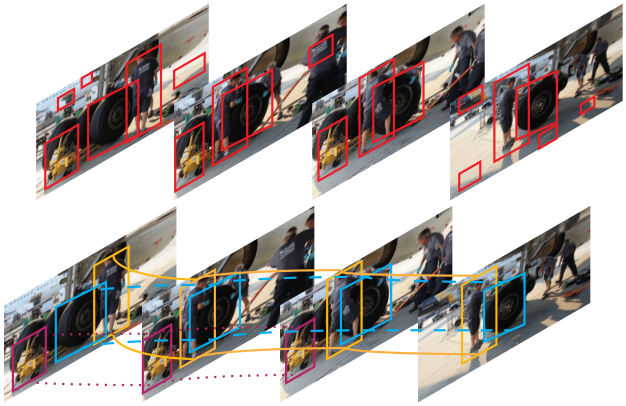


Figure 4. To overcome false positive detections from our object detector (top row), we exploit the spatiotemporal and appearance consistency among object proposals over time to link and prune them as its shown in the bottom row.

For each action proposal, we define $\mathbf{r}_i^t$ to denote the bounding box location of the i^{th} object proposal in time step t. We represent $\mathbf{r_i^t}$ with its ResNet object score vector, denoted as $\phi_o(\mathbf{r}_i^t) \in \mathbb{R}^o$. Unlike previous work that used global object scores at the frame/video level for action classification, we resort to object detection, so as to mitigate the effect of background in the representation and to fully exploit the action-object context. Due to the inherent challenges in video object detection [23, 42], we introduce a simple method to link object proposals over time, such that spurious and inconsistent detections do not contaminate the object-based representation of the action proposal (Figure 4 illustrates this step). Inspired by Gkioxari and Malik [12], we construct object tubes that have both spatiotemporal and appearance consistency. To form these tubes, we define the following linking score function:

$$l_s(\mathbf{r_i^t}, \mathbf{r_j^{t+1}})) = sim(\phi_o(\mathbf{r_i^t}), \phi_o(\mathbf{r_j^{t+1}})) + \lambda ov(\mathbf{r_i^t}, \mathbf{r_j^{t+1}})), \quad (1)$$

where $sim(\phi_o(\mathbf{r_i^t}), \phi_o(\mathbf{r_j^{t+1}}))$ is the cosine similarity between a pair of proposal object scores, and $ov(\mathbf{r_i^t}, \mathbf{r_j^{t+1}}))$ is the intersection over union between a pair of object proposal bounding boxes. Similar to [12], we cast the problem of finding the optimal path as:

$$\mathbf{R}^* = \arg\max_{\mathbf{R}} \frac{1}{T} \sum_{t=1}^{T-1} l_s(\mathbf{r_i^t}, \mathbf{r_j^{t+1}})), \quad (2)$$

for i and j in $\{1, \ldots, n_o\}$, where n_o is the total number of object proposals in a frame, and $\mathbf{R}$ is a sequence of linked object proposals. Equation 2 is efficiently solved using dynamic programming. We solve this problem at most N times, while removing the optimal path after every iteration. In practice, we set $N = 5$ in our experiments. This strategy allows us to generate object tubes with spatio-temporal appearance coherence.

Once object tubes are computed, we max pool the object responses of the object proposals in these tubes. To maintain temporal information, we use a temporal grid (of 16 bins in our experiments), within which max pooling is applied. Therefore, the object-level representation of an action proposal $\mathbf{p_i}$ is given by matrix $\mathbf{F}_o^i \in \mathbb{R}^{o \times 16}$.

Encoding action-scene relationships. Similar to objects, scenes have also demonstrated the ability to distinguish between human actions [29]. To encode scene information in an action proposal, we use a VGG network for large scale place recognition (VGGPlaces) [45], which is trained on the Places205 dataset [55]. For every proposal $\mathbf{p_i}$, we compute its VGGPlaces scene scores per time step. We max pool these scores 16 temporal bins spanning the whole action proposal uniformly. Therefore, the scene-level representation of an action proposal $\mathbf{p_i}$ is given by the matrix $\mathbf{F}_s^i \in \mathbb{R}^{s \times 16}$

Incorporating semantic context. Here, we aim to incorporate the semantic context available in $\mathbf{L}_o$ and $\mathbf{L}_s$ to prune action categories that are unlikely to be in action proposal $\mathbf{p}_i$. To do this, we simply enrich $\mathbf{p}_i$ with action-specific features according to Equation (3).

$$\psi_j^i = \begin{bmatrix} g(\mathbf{F}_o^i, \mathbf{L}_o(:,j)) \\ g(\mathbf{F}_s^i, \mathbf{L}_s(:,j)) \end{bmatrix} \forall i = 1, \ldots, n_p; \forall j = 1, \ldots, c \quad (3)$$

where $g(\mathbf{A}, \mathbf{b})$ performs an elementwise (Hadamard) vector product between each column of $\mathbf{A}$ and vector $\mathbf{b}$. In our case, $g(\mathbf{F}_o^i, \mathbf{L}_o(:,j))$ simply zeros out the effect of all object-level features corresponding to objects *not* associated with action j within the linking matrix $\mathbf{L}_o$. A similar idea holds for $g(\mathbf{F}_s^i, \mathbf{L}_s(:,j))$. In doing this, ψ_j^i can be viewed as the overall representation of the i^{th} action proposal for the j^{th} action class. For each action class j, we

train a 1-vs-all SVM classifier on the set of all ψ_j features to predict the action labels of proposals in our training set. As a cascade, this set of weak classifiers serves the purpose of reducing the number of false positives being fed into Stage 3. For each action proposal at testing time, we apply all these classifiers to compute c action confidence scores. In practice, we set a minimum threshold τ on these scores, so as to select a sparse number of action classes that are likely to be present in each proposal. Consequently, an action proposal $\mathbf{p}_i$, whose c action scores are less than τ is pruned from the original set of proposals. As such, m semantically consistent proposals remain from the original n_p and their thresholded scores are reserved in matrix $\mathbf{S}_{SCC} \in \{0,1\}^{c\times m}$.

In addition to giving each action proposal a class-specific score, we learn a regression function that fits the ψ_j features in the training set to the ground truth start and end locations of each proposal belonging to class j. Follow the parameterization of [11] but adjusted to temporal proposal, this class-specific regression function refines the location of proposal $\mathbf{p}_i$ based on c action categories. One regression model is learned per action category.At testing time, we only transform the action proposals for the classes selected in $\mathbf{S}_{SCC}$ to produce the updated class-specific locations $\mathbf{P}_{SCC}$.

4.3. Stage 3: Action Classifier

Much progress has been made in designing robust and highly accurate action classifiers [2, 6, 28, 38, 44, 46, 51]. So ideally, any of these classifiers can be used here. However, this would require sophisticated features to be extracted, which would significantly impact runtime. Alternatively, we reuse the visual representation used in Stage 1 (*i.e*. C3D features) and adopt the approach of Xu *et al*. [51] to build our final action classifier. The additional overhead in applying this classifier is minimal, as compared to using other more extravagant models.

To train this multi-class classifier, we augment the training ground-truth with action proposals, whose temporal intersection over union (tIoU) with the ground truth detections is greater than 0.7. Similarly, the set of negative examples is enlarged using action proposals with tIoU < 0.3. Here, we train linear SVM classifiers, using the C3D features encoded using VLAD [22].

At test time, we only apply the action classifiers selected in $\mathbf{S}_{SCC}$ at adjusted temporal locations $\mathbf{P}_{SCC}$. By this mean, the localization performance of our action classifier is boosted at a marginal cost depending on $\mathbf{S}_{SCC}$ as it is shown in Section 5. Finally, our localization results are further processed following standard practices such as non-maximum suppression (NMS) and multiply the detection scores by a class-specific length prior [33].

5. Experiments

Dataset. Traditional datasets for action detection [13, 54] contain only a small number of action categories (mostly sports), where the importance of semantic priors for large scale activity detection might not be fully appreciated. Recently, two large-scale datasets for video analysis [1, 24] were released to the vision community. Both datasets include activity/concept annotations at a global video level. Unfortunately, temporal boundaries of where the activities occur within the video are not available. Consequently, we choose to use ActivityNet [4], the largest available dataset for human activity analysis, in our experiments. Not only does ActivityNet include human activity annotations at the video level, but it also contains curated start and ending times of activities. These temporal annotations were generated based on a crowd-sourcing effort on Amazon Mechanical Turk. This dataset is quite diverse in terms of the type of activities too. For example, activities range from sports categories like *long jump* to household categories such as *Vacuuming floor*. In the last year, the authors released different versions of the dataset. In our experiments, we specifically use release 1.3 of ActivityNet which includes 200 activity classes and 19994 videos.

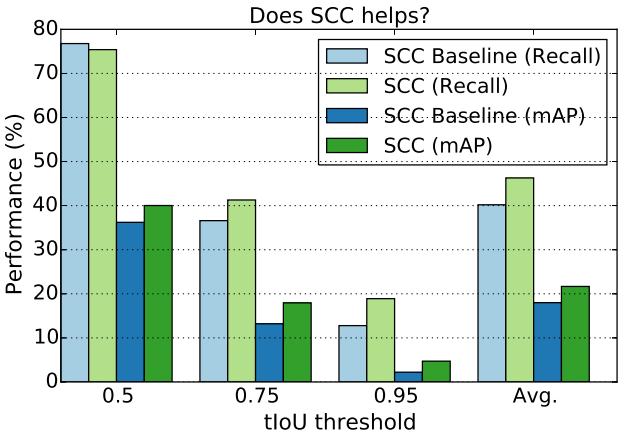
Implementation details. To make our findings reproducible, we describe here the implementation details of our SCC model. In our **action proposal** stage, we first extract DAPs proposals for the whole ActivityNet dataset. To filter out nearby action proposals, we apply non-maximum suppression with a tIoU threshold greater than 0.7. We reduce the total number of proposals per video by selecting only the top-100 scoring proposals. Our **semantic encoder** relies on EdgeBoxes [56] to extract the set of object proposals that feed our object tube module. Given that few objects are linked per activity, we limit the number of object tubes, N, to five per action proposal. Finally, our **action classifiers** are trained using a vocabulary of 512 k-means centers and the VLAD codes undergo power and L_2 normalization.

Baseline. Our baseline is a model that extracts the action proposals followed by the action classifier. In doing so, we detach the contribution of our SCC model. In other words, we define our baseline by turning off our semantic encoder. We refer to this approach as *SCC Baseline*.

Metrics. We follow the standard evaluation protocol in ActivityNet and compute the mean Average Precision (mAP) at different tIoU thresholds, *i.e*. 0.5, 0.75, 0.95, and average them from 0.5 to 0.95. To isolate the contribution of each of the early stages, we also report recall at the same tIoU thresholds used to compute mAP.

5.1. Experimental Analysis

To validate the contributions of our SCC model, we conduct a series of experiments evaluated on the validation set



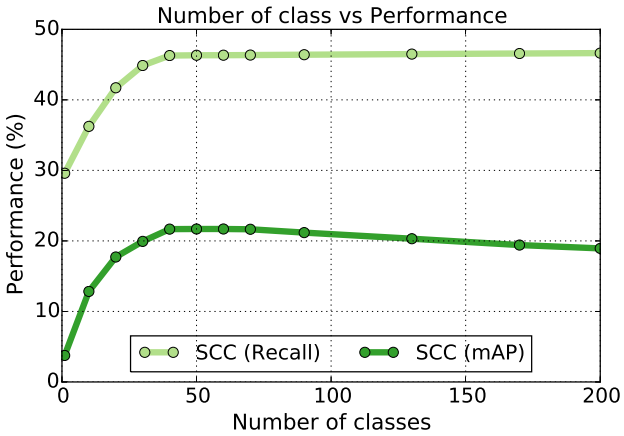


Figure 5. The left diagram shows the relevance of semantic context, in terms of gains in mAP and Recall, for temporal activity detection. On the right, we show the recall and detection performance in terms of the number of classifiers evaluated for each proposal at the last stage. In that order of ideas, we conclude that our SCC offers not only a efficient way to detect activities, it also mitigates the drop of performance by pruning out harmful actions for each segment.

of ActivityNet. We first compare the performance of SCC against our baseline. Then, we study several SCC variants with the end goal of isolating the contribution of each module in Stage 2 (semantic encoder).

Does SCC help? SCC significantly outperforms its baseline model (SCC Baseline) not only in terms of recall, but also in detection performance (mAP). Figure 5 (Left) compares both approaches in terms of recall and mAP at different tIoU thresholds. SCC achieves a large performance improvement at higher tIoU thresholds, which is attributed to SCC's ability to adjust temporal locations/scales when generating class-specific action proposals.

How many classes are fed to the action classifier? Not only does SCC generate high fidelity class-specific action proposals, it allows the selection of a sparse number of action classes to be fed to our action classifier. The sparsity is controlled by the minimum action score threshold τ. In Figure 5 (Right), we plot the mAP and recall of SCC with varying values of τ. When $\tau = 0$, all 200 action classes are fed to the classifier. Conversely, when τ increases, the number of classes that remain decreases. Interestingly, the recall and mAP of our method are not significantly affected when more than 75% of the classes are pruned out, thus, validating the importance of semantic context in the cascade. In fact, SCC achieves its highest performance when only 40 classes out of 200 are passed to the action classifier.

We investigate the factors that enable our SCC model to succeed. Table 1 compares the performance of different variants of our SCC model. Each variant is described and studied in-depth next.

Object tubes matter. We argue that globally encoding object scores deteriorates our action-object relationships. To demonstrate this, we report the performance of our SCC model when the object tubes are discarded. In other words, we obtain the object level representation $\mathbf{F}_o^i$ by max pooling over all the raw object proposal scores. As shown in Table 1, excluding object tubes (*w/o object tubes*) from our SCC model results in a significant drop in performance (recall and mAP). This highlights the ability of our object tubes to filter out noisy object detections.

	Recall (%)				mAP (%)			
SCC Variant	@50	@75	@95	@Avg	@50	@75	@95	@Avg
w/o object tubes	72.8	38.1	16.9	42.4	36.6	16.3	4.1	19.1
w/o regressor	72.5	34.8	15.9	41.9	39.8	15.9	3.1	19.9
w/o semantics	69.8	37.2	17.5	42.1	37.6	16.8	4.1	20.1
rnd semantics	40.3	29.6	10.7	30.5	29.1	10.0	1.7	10.7
full model	**75.4**	**41.3**	**18.9**	**46.3**	**40.0**	**17.9**	**4.7**	**21.7**

Table 1. Ablation study showing the relevance of all the components of our semantic encoder stage.

Proposal regression helps. When the class specific regression module (*w/o regressor*) is turned off, we observe that performance drastically decreases at higher tIoU thresholds (See Table 5). This is the case, since the class specific regression helps generate tighter segments, thus, translating into better performance at higher tIoU.

Semantic context. We define two different variants to unveil the importance of inducing semantic context into SCC and report results in Table 1. **(1)** We replace $\mathbf{L}_o$ and $\mathbf{L}_s$ with two randomly generated binary matrices (*rnd semantics*). **(2)** We replace $\mathbf{L}_o$ and $\mathbf{L}_s$ with two matrices of all ones. This is equivalent to connecting all objects and scenes to all actions (*w/o semantics*). As expected, performance decreases substantially when semantic context is replaced by randomly generated priors. This is an intuitive result due to the confusion introduced into the semantic encoder of SCC. A less drastic but still significant drop is observed for the *w/o semantics* variant. This verifies that using action-object and action-scene relationships mined from ActivityNet, ImageNet, and Places datasets improves the correctness of class-specific scores and regression results.

5.2. Comparison with State-of-the-Art

Table 2 compares SCC against state-of-the-art detection approaches on the ActivityNet testing set. It includes the detection performance at different tIoU thresholds, as well as, the runtime required at test time to process one minute of video. SCC consistently outperforms state-of-the-art approaches, when tighter predictions are desired (*i.e.* for tIoU greater than 0.5). In terms of speed, SCC reduces the computational cost 10 times, as compared to the fastest existing detection approach on ActivityNet release 1.3.

Although the *UTS Team* approach achieves the highest performance at tIoU of 0.5, it fails when stricter predictions (in terms of tIoU) are desired. Their approach strongly relies on duration and location biases on the dataset to produce candidate predictions, resulting in low performance at higher tIoU thresholds. Singh *et al.* [40] rely on expensive features to represent a sparse number of proposals. This approach obtains the second best performance (after SCC) when mAP is averaged over multiple tIoU thresholds. Singh *et al.* [39] also requires expensive optical flow measurements to describe the video sequence, but instead of using proposals, they rely on an LSTM to encode the temporal evolution of the video. This allows them to get competitive results over different tIoU thresholds. Finally, the *University of Tokyo* method uses cheap features to describe and then classify temporal segments generated using a sliding window approach. The cheap features allow them to reduce the computational cost, but at the price of losing important motion description. This results in overall lower performance as compared to other approaches. In terms of average mAP, SCC generally outperforms the state-of-the-art. For example, it registers a 1.5% improvement, as compared to the runner up. This improvement can be considered significant due to the difficulty of the task.

Another key property of our SCC model is that it detects actions in videos quickly. As compared to previous approaches, SCC is 10 times faster at testing time (see Table 2). SCC is able to scan and detect videos in real-time, which is desirable for large-scale scenarios.

	Performance (mAP%)				Test runtime	
Approach	@50	@75	@95	@Avg	Seconds	FPS
UTS Team	**42.5**	2.9	0.0	14.6	500	3.6
Singh *et al.* [40]	36.4	11.1	0.0	17.8	914	1.97
Singh *et al.* [39]	28.7	17.8	2.9	17.7	609	2.95
University of Tokyo	26.9	15.8	3.6	16.2	440	4.1
Our model	39.9	**18.7**	**4.7**	**19.3**	**50.2**	**35.9**

Table 2. Detection and Average Runtime performance in the test set of ActivityNet. Interestingly, SCC not only achieves state-of-art performance by exploiting the semantic context of activities, it is also the most efficient alternative among current approaches. Detailed runtime can be found in the supplementary material

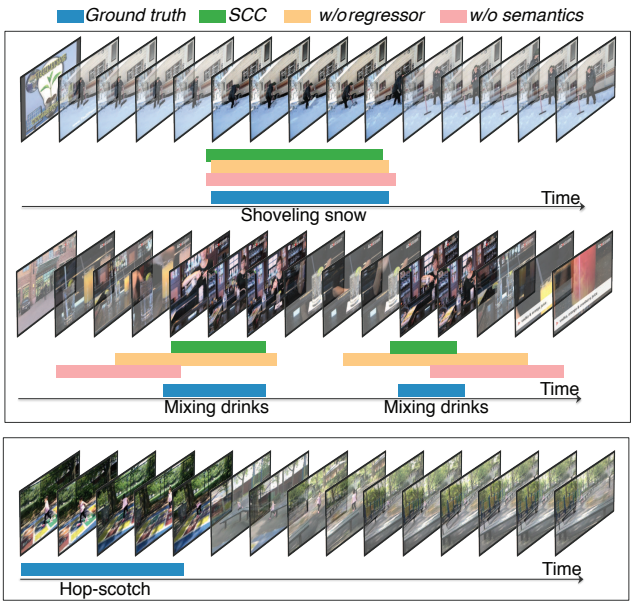


Figure 6. Qualitative results of different SCC variants. The first two rows show examples of videos where the right action is predicted. The last row shows a typical example where SCC fails.

5.3. Qualitative Results

Figure 6 shows qualitative results of different variants of SCC . Specifically, we present the detection results for the variants: *w/o regressor*, *w/o semantics* and our full model (SCC). The first two examples correspond to detections where all the approaches were able to predict the right class in the video. In the top example, all the variants accurately and tightly predict the action shoveling snow. However, for more difficult examples (as the second row), SCC outperforms the variants due to its ability to regress the locations of actions in a class-specific manner. Finally, the last row present an example where all the variants fail. As in this case, typical errors of SCC occurs when the intended activity does not include rich semantics. We include additional qualitative results in the supplementary material.

6. Conclusion

We introduce the Semantic Cascade Context (SCC) model, which is able to detect actions accurately and efficiently. SCC incorporated action-object and action-scene relationships with the end goal of improving recall of action proposals, while pruning out unrelated actions. Extensive experiments demonstrate that SCC produces robust detections and reduces the runtime at test time. In future work, we plan to explore how other vision tasks such as object detection can benefit from the mined semantic relationships.

Acknowledgments. Research in this publication was supported by the King Abdullah University of Science and Technology (KAUST) Office of Sponsored Research.

References

[1] S. Abu-El-Haija, N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan, and S. Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark. *arXiv preprint*, 2016.

[2] F. Basura, E. Gavves, J. M. Oramas, A. Ghodrati, and T. Tuytelaars. Modeling video evolution for action recognition. In *CVPR*, 2015.

[3] I. Biederman, R. J. Mezzanotte, and J. C. Rabinowitz. Scene perception: Detecting and judging objects undergoing relational violations. *Cognitive psychology*, 14(2):143–177, 1982.

[4] F. Caba Heilbron, V. Escorcia, B. Ghanem, and J. Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, 2015.

[5] F. Caba Heilbron, J. C. Niebles, and B. Ghanem. Fast temporal activity proposals for efficitent detection of human actions in untrimmed videos. In *CVPR*, 2016.

[6] F. Caba Heilbron, A. Thabet, J. C. Niebles, and B. Ghanem. Camera motion and surrounding scene appearance as context for action recognition. In *ACCV*, 2014.

[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[8] O. Duchenne, I. Laptev, J. Sivic, F. Bach, and J. Ponce. Automatic annotation of human actions in video. In *ICCV*, 2009.

[9] V. Escorcia, F. Caba Heilbron, J. C. Niebles, and B. Ghanem. Daps: Deep action proposals for action understanding. In *ECCV*, 2016.

[10] A. Gaidon, Z. Harchaoui, and C. Schmid. Actom sequence models for efficient action detection. In *CVPR*, 2011.

[11] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.

[12] G. Gkioxari and J. Malik. Finding action tubes. In *CVPR*, 2015.

[13] A. Gorban, H. Idrees, Y.-G. Jiang, A. Roshan Zamir, I. Laptev, M. Shah, and R. Sukthankar. THUMOS challenge: Action recognition with a large number of classes. http://www.thumos.info/, 2015.

[14] A. Gupta and L. S. Davis. Objects in action: An approach for combining action understanding and object perception. In *CVPR*, 2007.

[15] P. M. S. Hacker. Events and objects in space and time. *Mind*, 91(361):1–19, 1982.

[16] D. Han, L. Bo, and C. Sminchisescu. Selection and context for action recognition. In *ICCV*, 2009.

[17] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[18] J. Hosang, R. Benenson, P. Dollár, and B. Schiele. What makes for effective detection proposals? *IEEE transactions on pattern analysis and machine intelligence*, 38(4):814–830, 2016.

[19] N. Ikizler-Cinbis and S. Sclaroff. Object, scene and actions: Combining multiple features for human action recognition. In *ECCV*, 2010.

[20] M. Jain, J. C. van Gemert, T. Mensink, and C. G. Snoek. Objects2action: Classifying and localizing actions without any video example. In *ICCV*, 2015.

[21] M. Jain, J. C. van Gemert, and C. G. Snoek. What do 15,000 object categories tell us about classifying and localizing actions? In *CVPR*, 2015.

[22] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, and C. Schmid. Aggregating local image descriptors into compact codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(9):1704–1716, 2012.

[23] K. Kang, W. Ouyang, H. Li, and X. Wang. Object detection from video tubelets with convolutional neural networks. In *CVPR*, 2016.

[24] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014.

[25] H. S. Koppula, R. Gupta, and A. Saxena. Learning human activities and object affordances from rgb-d videos. *The International Journal of Robotics Research*, 32(8):951–970, 2013.

[26] L.-J. Li and L. Fei-Fei. What, where and who? classifying events by scene and object recognition. In *ICCV*, 2007.

[27] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: common objects in context. *arXiv preprint*, 2014.

[28] S. Ma, L. Sigal, and S. Sclaroff. Learning activity progression in lstms for activity detection and early detection. In *CVPR*, 2016.

[29] M. Marszalek, I. Laptev, and C. Schmid. Actions in context. In *CVPR*, 2009.

[30] P. Mettes, J. C. van Gemert, S. Cappallo, T. Mensink, and C. G. Snoek. Bag-of-fragments: Selecting and encoding video fragments for event detection and recounting. In *ICMR*, 2015.

[31] A. Montes, A. Salvador, and X. Giró i Nieto. Temporal activity detection in untrimmed videos with recurrent neural networks. *arXiv preprint*, 2016.

[32] D. Oneata, J. Verbeek, and C. Schmid. Efficient action localization with approximately normalized fisher vectors. In *CVPR*, 2014.

[33] D. Oneata, J. Verbeek, and C. Schmid. The lear submission at thumos 2014. 2014.

[34] A. Prest, V. Ferrari, and C. Schmid. Explicit modeling of human-object interactions in realistic videos. *IEEE transactions on pattern analysis and machine intelligence*, 35(4):835–848, 2013.

[35] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2015.

[36] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[37] Z. Shou, D. Wang, and S.-F. Chang. Temporal action localization in untrimmed videos via multi-stage cnns. In *CVPR*, 2016.

[38] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 2014.

[39] B. Singh, T. K. Marks, M. Jones, O. Tuzel, and M. Shao. A multi-stream bi-directional recurrent neural network for fine-grained action detection. In *CVPR*, 2016.

[40] G. Singh and F. Cuzzolin. Untrimmed video classification for activity detection: submission to activitynet challenge. *arXiv preprint*, 2016.

[41] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015.

[42] S. Tripathi, Z. C. Lipton, S. J. Belongie, and T. Q. Nguyen. Context matters: Refining object detection in video with recurrent neural networks. In *BMVC*, 2016.

[43] J. van Gemert, M. Jain, E. Gati, and C. G. M. Snoek. APT: Action localization proposals from dense trajectories. In *BMVC*, 2015.

[44] H. Wang and C. Schmid. Action recognition with improved trajectories. In *ICCV*, 2013.

[45] L. Wang, S. Guo, W. Huang, and Y. Qiao. Places205-vggnet models for scene recognition. *arXiv preprint*, 2015.

[46] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Val Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, 2016.

[47] P. Wei, Y. Zhao, N. Zheng, and S. C. Zhu. Modeling 4d human-object interactions for joint event segmentation, recognition, and object localization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP(99):1–1, 2016.

[48] P. Weinzaepfel, Z. Harchaoui, and C. Schmid. Learning to track for spatio-temporal action localization. In *ICCV*, 2015.

[49] J. Wu, A. Osuntogun, T. Choudhury, M. Philipose, and J. M. Rehg. A scalable approach to activity recognition based on object use. In *ICCV*, 2007.

[50] Z. Wu, Y. Fu, Y.-G. Jiang, and L. Sigal. Harnessing object and scene semantics for large-scale video understanding. In *CVPR*, 2016.

[51] Z. Xu, Y. Yang, and A. G. Hauptmann. A discriminative cnn video representation for event detection. In *CVPR*, 2015.

[52] S. Yeung, O. Russakovsky, G. Mori, and L. Fei-Fei. End-to-end learning of action detection from frame glimpses in videos. In *CVPR*, 2016.

[53] G. Yu and J. Yuan. Fast action proposals for human action detection and search. In *CVPR*, 2015.

[54] J. Yuan, Z. Liu, and Y. Wu. Discriminative subvolume search for efficient action detection. In *CVPR*, 2009.

[55] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning deep features for scene recognition using places database. In *NIPS*, 2014.

[56] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In *ECCV*, 2014.